RS001R.MBR

Path: NXCLOUD/rpgsrc/RS001R.MBR **Generated**: 2026-01-08 13:32:57 **Processing Time**: 12144ms

Business Logic for ASOKON: Kontroll av sist brukte kundenummer

This document outlines the business rules that govern the retrieval and updating of the last used customer number in the ASOKON system, based on an analysis of the RPG program RS001R. The primary focus is on the logic that determines how the last used customer number is fetched, validated, and updated.

The core logic for retrieving the last used customer number is contained within the main processing logic of the RS001R program. The program checks for the existence of the last used customer number in various files, applies business rules to determine the next available customer number, and updates the relevant records accordingly.

Customer Number Retrieval and Validation Rules

ASOKON: RAA4PFR, RAA1PFR, RkunPFR

1. Retrieve Last Used Customer Number

- **Logic:** The program retrieves the last used customer number from the RAA4PFR file. If the record is not found, it sets a return code to indicate failure.
- **File:** RAA4PFR (Last used customer number file)
- **Field:** RA4KNR
- **Condition:** The process will not select a record if the RAA4L1 record is not found (indicated by *IN90 being on).

2. Determine Next Customer Number

- **Logic:** Based on the retrieved customer number, the program calculates the next available customer number based on specific conditions (incrementing or decrementing).
- **File:** RAA4PFR (Last used customer number file)
- **Field:** RA4KNR
- **Condition:** If the w_test variable is blank, increment the customer number. If w_test is 'S', decrement if it matches the last used number. If w_test is 'K', set it to the parameter customer number.

Boundary Check Rules

1. Check Customer Number Against Limits

- **Logic:** The program checks if the calculated customer number is within defined limits (minimum and maximum).
- **Files:**
- RAA1PFR (Boundary values file)
- **Fields:**
- RA1HKT (Lower boundary)
- RA1HRS (Upper boundary)

- **Condition:** The process will set the return code to 'N' if the calculated customer number is less than or equal to RA1HKT or greater than or equal to RA1HRS.

Customer Number Availability Rules

1. Check if Customer Number Exists

- **Logic:** The program checks if the calculated customer number exists in the customer register. If it does, it increments the number until an available one is found.
- **File:** RkunPFR (Customer register file)
- **Field:** RKKUND
- **Condition:** The process continues to check until an available customer number is found (indicated by *IN90 being off).

2. Update Last Used Customer Number

- **Logic:** If a new customer number is found, the program updates the last used customer number in the RAA4PFR file.
- **File:** RAA4PFR (Last used customer number file)
- **Field:** RA4KNR
- **Condition:** The update occurs if the customer number found is greater than the last used number.

Special Conditions (Program-Specific)

1. Initialization and Parameter Handling (RS001R)

- **Logic:** The program initializes parameters passed from the calling program and sets up keys for file access.
- **File:** Various files including RAA1PFR, RAA4PFR, and RkunPFR
- **Fields:** w_firm, p_4knr
- **Condition:** The program must successfully retrieve boundary values before proceeding with customer number logic.

Subprogram Calls Affecting Logic

Beyond direct file checks, several external subprograms are called that play a significant role in the workflow.

1. RS001R (Last Used Customer Number Control)

- **Trigger:** This program is called to retrieve and update the last used customer number.
- **Logic:** It checks for the last used customer number, validates it against business rules, and updates the record if necessary.
- **Impact:** This call acts as a **critical control point** for managing customer numbers in the system.

2. RAA1L1 (Boundary Values Retrieval)

- **Trigger:** This program is called to fetch boundary values for customer number limits.
- **Logic:** It retrieves minimum and maximum values for customer numbers.
- **Impact:** This is essential for ensuring that customer numbers remain within acceptable limits.

3. RkunLR (Customer Register Check)

- **Trigger:** This program is called to verify the existence of a customer number in the customer register.

- **Logic:** It checks if the calculated customer number is already in use.
- **Impact:** This represents a **key validation step** in the customer number assignment process, ensuring uniqueness.