RS001R.MBR

Path: NXCLOUD/rpgsrc/RS001R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 11242ms

Business Logic for Customer Number Retrieval

This document outlines the business rules that govern the retrieval of the last used customer number, based on an analysis of the RPG program RS001R. The primary focus is on how the program processes and updates customer numbers based on specific conditions.

The core logic for retrieving the last used customer number is contained within the main program logic of RS001R. The program checks various conditions to determine the appropriate customer number to return and updates the records accordingly.

Customer Number and Record Rules

Customer Number Retrieval: raa4pfr, rkunpfr, raa1pfr

1. Check for Last Used Customer Number

- **Logic:** The program attempts to retrieve the last used customer number from the customer register. If not found, it sets a return flag to indicate failure.
- **File:** raa4pfr (Last used customer number file)
- **Field:** ra4knr
- **Condition:** The process will not select a record if the last used customer number cannot be found (indicated by *in90 being on).

2. Update Last Used Customer Number

- **Logic:** If the last used customer number is found, the program updates it based on specific logic that considers the state of the w_test variable.
- **File:** raa4pfr (Last used customer number file)
- **Field:** ra4knr
- **Condition:** The last used customer number is updated only if it is within defined limits (ra1hkt and ra1hrs).

Configuration and Boundary Rules

1. Boundary Check for Customer Number

- **Logic:** The program checks if the calculated customer number (w_4knr) is within the defined boundaries. If it is outside these boundaries, it sets a return flag to indicate failure.
- **Files:**
- raa1pfr (Boundary values file)
- **Fields:**
- ra1hkt (Lower boundary)
- ra1hrs (Upper boundary)
- **Condition:** The process will set p_retk to 'N' if w_4knr is less than or equal to ra1hkt or greater than or equal to ra1hrs.

2. Check for Existing Customer Number

- **Logic:** The program checks if the calculated customer number exists in the customer register. If it does, it increments the customer number until a free number is found.

- **File:** rkunpfr (Customer register)
- **Field:** rkkund
- **Condition:** The process will continue to increment w_kund until a number is found that does not exist in the customer register.

Special Conditions (Program-Specific)

1. Return Status Code (RS001R)

- **Logic:** The program sets a return status code based on the success or failure of the customer number retrieval process.
- **File:** raa4pfr (Last used customer number file)
- **Field:** p_retk
- **Condition:** The return code is set to 'N' if the customer number is not found or if it is outside the defined boundaries.

2. Update Last Used Customer Number (RS001R)

- **Logic:** The program updates the last used customer number in the file if a valid number is found.
- **File:** raa4pfr (Last used customer number file)
- **Fields:** ra4knr (Last used customer number)
- **Condition:** The update occurs only if the customer number is greater than the current last used number or if specific conditions regarding w_test are met.

Subprogram Calls Affecting Logic

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

1. RS001R (Customer Number Retrieval Program)

- **Trigger:** This program is called to retrieve and update the last used customer number.
- **Logic:** It checks for the last used customer number, updates it based on specific conditions, and checks against boundaries.
- **Impact:** This call acts as a **major logical gateway** for managing customer numbers within the system.

2. RAA1L1 (Boundary Value Retrieval)

- **Trigger:** This program is called to retrieve boundary values for customer numbers.
- **Logic:** It fetches the minimum and maximum values that customer numbers can take.
- **Impact:** This is a **critical validation step** that ensures customer numbers remain within acceptable limits.

3. RkunLR (Customer Existence Check)

- **Trigger:** This program is called to check if a customer number already exists in the register.
- **Logic:** It verifies the existence of the customer number and helps in determining the next available number.
- **Impact:** This represents the handoff to the next major business function, ensuring unique customer number assignment.