RA001R.MBR

Path: NXCLOUD/rpgsrc/RA001R.MBR **Generated**: 2026-01-08 15:43:00 **Processing Time**: 20781ms

Business Logic for Vedlikehold av statuskoder

This document outlines the business rules that govern the maintenance of status codes 1 and 3, based on an analysis of the RPG program RA001R. The primary focus is on how status codes are retrieved, validated, and updated within the system.

The core logic for maintaining status codes is contained within the A2BLD subroutine in RA001R. The program retrieves existing status codes, validates the input data against business rules, and updates the records accordingly.

Order Status and Header Rules

Vedlikehold av statuskoder: raa1l1, raa3l1

1. Retrieve Status Code 1

- **Logic:** The program retrieves the status code 1 from the raa1l1 file. If the record is found, it populates various fields with default values; otherwise, it retains the existing values.
- **File:** raa1l1 (Status Code 1 File)
- **Field:** ra1ape, ra1per, ra1aar, ra1hkt, ra1hrs, etc.
- **Condition:** The process will not select a record if the chain operation fails (indicated by *in60 being off).

2. Retrieve Status Code 3

- **Logic:** The program retrieves the status code 3 from the raa3l1 file. Similar to status code 1, it populates fields based on the retrieved record.
- **File:** raa3l1 (Status Code 3 File)
- **Field:** ra3stp, ra3p01, ra3p02, etc.
- **Condition:** The process will not select a record if the chain operation fails (indicated by *in61 being off).

Configuration and Authorization Rules

1. Firm Association

- **Logic:** The program associates the current firm number with the status codes being processed.
- **Files:**
- raa1l1 (Status Code 1 File)
- raa3l1 (Status Code 3 File)
- **Fields:**
- w_firm (from the working storage)
- ra1fir (from the status code 1 file)
- **Condition:** The firm number must be set before processing status codes.

2. Exit Conditions

- **Logic:** The program checks for exit conditions based on user input (F1 for selection, F3 for exit).
- **File:** None

•**Field:** None

•**Condition:** The program will exit if the user selects the exit option.

Financial and Transactional Rules

1. Validation of Periods

•**Logic:** The program checks that the period number (a21ape) is within valid limits (1 to 13).

•**File:** None

•**Fields:**

•a21ape (Current period)

•**Condition:** The process will trigger an error if a21ape is less than 1 or greater than 13.

2. Year Validation

•**Logic:** The program checks that the year (a21aar) is not less than 1980 if a period is specified.

•**File:** None

•**Condition:** The process will trigger an error if a21aar is less than 1980 when a21per is greater than zero.

Special Conditions (Program-Specific)

1. Update Status Code 1 (RA001R)

•**Logic:** The program updates the fields of status code 1 based on user input and current values.

•**File:** raa1lu (Update Status Code 1 File)

•**Field:** ra1ape, ra1per, ra1aar, etc.

•**Condition:** The update occurs if the chain operation is successful; otherwise, a write operation is performed.

2. Update Status Code 3 (RA001R)

•**Logic:** Similar to status code 1, the program updates the fields of status code 3 based on user input.

•**File:** raa3lu (Update Status Code 3 File)

•**Fields:** ra3stp, ra3p01, ra3p02, etc.

•**Condition:** The update occurs if the chain operation is successful; otherwise, a write operation is performed.

Subprogram Calls Affecting Logic

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

1. A2BLD (Screen for Status Code Maintenance)

•**Trigger:** This program is called to display the maintenance screen for status codes.

•**Logic:** It handles the input and output of status codes, ensuring that the user can view and modify them.

•**Impact:** This call acts as a **major logical gateway** for user interaction with status codes.

2. keya1 and keya3 (Key Lists for Status Codes)

•**Trigger:** These key lists are initialized at the beginning of the program to facilitate record retrieval.

- **Logic:** They define the keys used for accessing status code records.
- **Impact:** This is essential for ensuring that the correct records are accessed and modified.

3. dummy (Tag for Flow Control)

- **Trigger:** This tag is used to manage flow control within the program.
- **Logic:** It serves as a placeholder for branching logic.
- **Impact:** Helps maintain the program's structure and flow, ensuring proper execution of business rules.