

RA001R.MBR

Path: NXCLOUD/rpgsrc/RA001R.MBR **Generated:** 2026-01-08 12:35:07 **Processing Time:** 12057ms

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 processes records for status codes 1 and 3, performing validations and updates based on user input.

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 the corresponding values.
- File:** raa1l1 (Status Code 1 File)
- Field:** ra1ape, ra1per, ra1aar, ra1hkt, ra1hrs, etc.
- Condition:** The process will not select a record if the status code does not exist (indicated by *in60 being off).

2. Retrieve Status Code 3

- Logic:** The program retrieves the status code 3 from the raa3l1 file. If the record is found, it populates various fields with the corresponding values.
- File:** raa3l1 (Status Code 3 File)
- Field:** ra3stp, ra3p01, ra3p02, etc.
- Condition:** The process will not select a record if the status code does not exist (indicated by *in61 being off).

Configuration and Authorization Rules

1. Firm Validation

- Logic:** The program checks if the firm number is valid before processing the status codes.
- Files:**
 - raa1l1 (Status Code 1 File)
 - raa3l1 (Status Code 3 File)
- Fields:**
 - w_firm (Firm Number)
- Condition:** The firm number must match the key fields for both status codes.

2. Exit Conditions

- Logic:** The program allows users to exit the maintenance screen using specific function keys.
- File:** a2bld (Maintenance Screen)
- Field:** None

- Condition:** The program exits if the user presses F3 or other designated exit keys.
-

Financial and Transactional Rules

1. Period Validation

- Logic:** The program checks if the period entered is within the acceptable range (1-13).
- File:** raa1l1 (Status Code 1 File)
- Fields:**
 - a21ape (Entered 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 if the year is valid and greater than 1980.
 - File:** raa1l1 (Status Code 1 File)
 - Condition:** The process will trigger an error if the year (a21aar) is less than 1980 and the period (a21per) is greater than zero.
-

Special Conditions (Program-Specific)

1. Update Status Code 1 (RA001R)

- Logic:** The program updates the status code 1 record with the values from the maintenance screen.
- File:** raa1lu (Status Code 1 Update File)
- Field:** ra1ape, ra1per, ra1aar, etc.
- Condition:** The update occurs if *in60 is off; otherwise, it writes a new record.

2. Update Status Code 3 (RA001R)

- Logic:** The program updates the status code 3 record with the values from the maintenance screen.
 - File:** raa3lu (Status Code 3 Update File)
 - Fields:** ra3stp, ra3p01, ra3p02, etc.
 - Condition:** The update occurs if *in61 is off; otherwise, it writes a new record.
-

Subprogram Calls Affecting Logic

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

1. A2BLD (Maintenance Screen for Status Codes)

- Trigger:** This program is called to display the maintenance screen for status codes.
- Logic:** It handles the retrieval and display of status codes and user input for updates.
- Impact:** This call acts as a **major logical gateway** for maintaining status codes.

2. keya1 and keya3 (Key Definitions)

- Trigger:** These keys are defined at the beginning of the program to manage access to status code records.
- Logic:** They ensure that the correct firm number is used when retrieving or updating records.
- Impact:** This is a **critical control mechanism** for ensuring data integrity.

3. *inzsr (Initialization Subroutine)

- Trigger:** This subroutine initializes the program and sets up key fields.
- Logic:** It populates the firm number and prepares the keys for status code retrieval.
- Impact:** This represents the handoff to the next major business function of retrieving and updating status codes.