SX905R.MBR

Path: NXCLOUD/rpgsrc/SX905R.MBR **Generated**: 2026-01-09 10:13:45 **Processing Time**: 11598ms

Business Logic for Vedlikehold av betingelser for omregning

This document outlines the business rules that govern the maintenance of logs and status transfers for Cobuilder, based on an analysis of the RPG program SX905R. The primary focus is on how the program interacts with status records and manages user inputs.

The core logic for maintaining the status records is contained within the a1taga subroutine in SX905R. The program retrieves information from the status register, processes user inputs, and updates the records accordingly.

Order Status and Header Rules

Vedlikehold av logg/status overføring Cobuilder: scobl1, scoblu

1. Retrieve Status Information

- **Logic:** The program retrieves status information from the scobl1 file. If a record is found, it populates various fields with the corresponding data; if not found, it initializes the fields to default values.
- **File:** scobl1 (Status register)
- **Field:** scstat, scmai1, scmai2, scmai3, scsubj, sctxt1, sctxt2, sctxt3, scsend
- **Condition:** The process will not select a record if scobl1_key is not found.

2. Check Email Address

- **Logic:** The program checks if the primary email address (a1mai1) is blank. If it is, an error indicator is set.
- **File:** scobl1 (Status register)
- **Field:** a1mai1
- **Condition:** The process will trigger an error if a1mai1 is equal to *blanks.

Configuration and Authorization Rules

1. Update Status Record

- **Logic:** The program updates the status record in scoblu based on the values entered by the user. If the record is found, it updates the existing record; if not, it creates a new record.
- **Files:**
- scoblu (Update status register)
- scobl1 (Status register)
- **Fields:**
- scstat (from scoblu)
- scmai1 (from scoblu)
- **Condition:** The update occurs if the record is found in scoblu; otherwise, a new record is created with the firm number.

2. Check Status Validity

•**Logic:** The program verifies that the status (a1stat) is either 0 or 1 before proceeding. If it is not, an error indicator is set.
•**File:** scobl1 (Status register)
•**Field:** a1stat
•**Condition:** The process will trigger an error if a1stat is not equal to 0 or 1.

Financial and Transactional Rules

1. Log User Actions

•**Logic:** The program logs the user who made changes to the status record by capturing the user ID and timestamps.
•**File:** scoblu (Update status register)
•**Fields:**
•sceusr (User ID)
•scedat (Date)
•scetim (Time)
•**Condition:** The logging occurs upon updating the status record.

2. Handle Print Request

•**Logic:** If the user requests to print (*inkf), the program calls a subroutine to handle the print operation.
•**File:** scobl1 (Status register)
•**Condition:** The print request is processed if *inkf is equal to *on.

Special Conditions (Program-Specific)

1. Delete Marked Messages (SX903R)

•**Logic:** The program calls another program (SX903R) to delete updated or marked messages based on user input.
•**File:** scobl1 (Status register)
•**Field:** w_dato
•**Condition:** The deletion process is initiated based on user confirmation.

2. Initialize Program Variables

•**Logic:** The program initializes key variables and sets the firm number for subsequent operations.
•**File:** scobl1 (Status register)
•**Fields:** w_firm (Firm number)
•**Condition:** This initialization occurs at the start of the program.

Subprogram Calls Affecting Logic

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

1. SX903R (Delete Messages Program)

•**Trigger:** Called when the user opts to delete marked messages.
•**Logic:** This program handles the deletion of messages based on the parameters passed from SX905R.

•**Impact:** This call represents the handoff to the next major business function of message management.

2. xb6win (Print Handling Subroutine)

•**Trigger:** Invoked when the user requests to start printing.

•**Logic:** This subroutine manages the print operation and user confirmations.

•**Impact:** This is a critical step for user interaction and document generation.

3. *inzsr (Initialization Subroutine)

•**Trigger:** Automatically called at the start of the program.

•**Logic:** Initializes keys for looking up records in the status register.

•**Impact:** This establishes the necessary context for the program's operations.