SX905R.MBR

Path: NXCLOUD/rpgsrc/SX905R.MBR **Generated**: 2026-01-08 13:33:09 **Processing Time**: 14496ms

Business Logic for Vedlikehold av betingelser for omregning

This document outlines the business rules that govern the maintenance of logs/status for the transfer of Cobuilder, based on an analysis of the RPG program SX905R. The primary focus is on how the program retrieves, updates, and manages status information.

The core logic for managing the status is contained within the main processing logic of the program, which handles the retrieval of information from the status register and updates it based on user input and conditions.

Order Status and Header Rules

Vedlikehold av logg/status overføring Cobuilder: scobl1, scoblu

1. Retrieve Status Information

- **Logic:** The program retrieves information from the status register (scobl1) based on a key. If found, it populates various fields with the retrieved data; if not found, it initializes these fields to default values.
- **File:** scobl1 (Status Register)
- **Field:** scstat, scmai1, scmai2, scmai3, scsubj, sctxt1, sctxt2, sctxt3, scsend, sckdat, scudat, scutid
- **Condition:** The process will not select a record if the key does not match any existing records in scobl1.

2. Display Status Information

- **Logic:** The program displays the retrieved status information on the screen for user interaction.
- **File:** x905d (Workstation Display)
- **Field:** Various fields displayed on the screen
- **Condition:** The display is triggered after retrieving the status information.

Configuration and Authorization Rules

1. Check for Valid Status

- **Logic:** The program checks if the status is valid (not equal to 0 or 1). If invalid, it sets an error indicator.
- **File:** x905d (Workstation Display)
- **Field:** a1stat
- **Condition:** The check occurs if the status is neither 0 nor 1.

2. Validate Email Address

- **Logic:** The program checks if the primary email address field is blank. If it is, it sets an error indicator.
- **File:** x905d (Workstation Display)
- **Field:** a1mai1

•**Condition:** The check is performed if a1mai1 is blank.

Financial and Transactional Rules

1. Update Status Register

•**Logic:** The program updates the status register (scoblu) with the current values from the user input. If the record is found, it updates the existing record; if not found, it creates a new record.

•**File:** scoblu (Updated Status Register)

•**Fields:**

•scstat (Status)

•scmai1 (Email 1)

•scmai2 (Email 2)

•scmai3 (Email 3)

•scsubj (Subject)

•sctxt1 (Text 1)

•sctxt2 (Text 2)

•sctxt3 (Text 3)

•scsend (Send Flag)

•**Condition:** The update occurs if the record is found; otherwise, a new record is created.

2. Log User Activity

•**Logic:** The program logs the user and timestamp for the update operation.

•**File:** scoblu (Updated Status Register)

•**Condition:** The logging happens during the update process.

Special Conditions (Program-Specific)

1. Subroutine for Deleting Messages (SX903R)

•**Logic:** This subroutine handles the deletion of updated and/or marked messages based on user input.

•**File:** f6win (Window for Deletion)

•**Field:** b1kdat (Date for deletion)

•**Condition:** The subroutine is called if the user indicates a desire to delete messages.

2. Subroutine for Initialization

•**Logic:** This subroutine initializes the program, setting up keys for lookups in the status register and populating the firm number.

•**File:** scobl1, scoblu (Status Registers)

•**Fields:** w_firm (Firm Number)

•**Condition:** This subroutine is executed 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 (Message Deletion Program)

•**Trigger:** Called from the deletion subroutine when the user opts to delete messages.

•**Logic:** This program processes the deletion of messages based on the provided date and inclusion flag.
•**Impact:** This call represents a **critical step in managing message data** and ensures that the system remains clean of unnecessary records.

2. *inzsr (Initialization Subroutine)

•**Trigger:** Automatically called at the beginning of the program.
•**Logic:** Initializes keys for accessing the status register and sets the firm number.
•**Impact:** This ensures that the program has the necessary context to perform its operations correctly.

3. exfmt (Display Format Subroutine)

•**Trigger:** Called to display the status information on the screen.
•**Logic:** Handles the formatting and presentation of data to the user.
•**Impact:** This is a **key interaction point** for user engagement with the status information.