SX905R.MBR

Path: NXCLOUD/rpgsrc/SX905R.MBR **Generated**: 2026-01-08 12:59:12 **Processing Time**: 15177ms

Business Logic for Vedlikehold av betingelser for omregning

This document outlines the business rules that govern the maintenance of logs and status for the Cobuilder transfer process, based on an analysis of the RPG program SX905R. The primary focus is on how the program manages the retrieval and updating of status records.

The core logic for maintaining status records is contained within the A1BLD subroutine in SX905R . The program processes records from the scobl1 and scoblu files, checking for existing entries and updating or inserting new records as necessary.

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 matching key is found. If found, it populates various status fields; if not found, it initializes the 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 entry in the scobl1 file.

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 Register

- **Logic:** The program updates the scoblu file with the current status and user information. If the record exists, it updates; if not, it creates a new entry.
- **Files:**
- scoblu (Update 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. Check Status Validity

•**Logic:** The program checks if the status is valid (not 0 or 1). If invalid, it sets an indicator to show an error.
•**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. Initiate Program

•**Logic:** The program initializes by setting up keys for the status register based on the firm number.
•**File:** scobl1, scoblu (Status registers)
•**Fields:**
•w_firm (Firm number)
•**Condition:** The keys are set based on the firm number provided in the local data area.

2. Print Status Report

•**Logic:** If the print flag is set, the program calls a subroutine to handle the printing of the status report.
•**File:** scobl1 (Status register)
•**Condition:** The print action is triggered if the print flag (*inkf) is on.

Special Conditions (Program-Specific)

1. Delete Updated 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:** This action is performed if the user indicates a desire to delete messages.

2. Error Handling for User Input

•**Logic:** The program checks for specific user input conditions and sets error indicators accordingly.
•**File:** scobl1 (Status register)
•**Fields:** *in30, *in32 (Error indicators)
•**Condition:** The program will set error indicators based on user input flags.

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 during the xb6win subroutine when the user opts to delete messages.
- **Logic:** This program handles the deletion of messages based on the date provided by the user.
- **Impact:** This call acts as a **clean-up step** to ensure that outdated or unnecessary messages are removed from the system.

2. exfmt (Display Format)

- **Trigger:** Called to display the status form to the user.
- **Logic:** This function displays the current status and allows user interaction.
- **Impact:** This is a **critical user interface step** that facilitates user interaction with the status records.

3. *inzsr (Initialization Subroutine)

- **Trigger:** Automatically called at the start of the program to set up initial conditions.
- **Logic:** Initializes keys and sets up the necessary environment for processing.
- **Impact:** This represents the **initial setup phase** of the program, ensuring that all necessary data is ready for processing.