RP001R.MBR

Path: NXCLOUD/rpgsrc/RP001R.MBR **Generated**: 2026-01-08 15:43:00 **Processing Time**: 17681ms

Business Logic for Koder Vedlikehold

This document outlines the business rules that govern the maintenance of codes in the ASOKON system, based on an analysis of the RPG program RP001R. The primary focus is on how the program handles code types, specifically "Prosjektkategori" (Project Category).

The core logic for maintaining codes is contained within the various subroutines in RP001R. The program utilizes subfiles to manage the display and interaction with code records, allowing for operations such as create, update, delete, and navigation through the records.

Order Status and Header Rules

Koder Vedlikehold: frpkol1, frpkol2, frpkolr, frpkolu

1. Record Selection

- **Logic:** The program will select records from the frpkol1 and frpkol2 files based on user input and the current state of the subfile.
- **File:** frpkol1 (Project Category Codes)
- **Field:** rpkol1_kkod
- **Condition:** Records are selected if the user has provided a valid code or description in the input fields.

2. Subfile Display

- **Logic:** The program displays the current records in a subfile format, allowing users to navigate through the list of codes.
- **File:** frpkol1 (Project Category Codes)
- **Field:** b1sfl
- **Condition:** The subfile will be populated if there are records available to display.

Configuration and Authorization Rules

1. User Initialization

- **Logic:** The program initializes user-specific variables and sets the firm and type based on the local data area.
- **Files:**
- frpkol1 (Project Category Codes)
- frpkol2 (Project Category Descriptions)
- **Fields:**
- l_firm (Firm Identifier)
- p_type (Code Type)
- **Condition:** The initialization occurs at the start of the program when the user enters the program.

2. Function Key Handling

- **Logic:** The program handles function keys for various operations such as creating, updating, and deleting records.

•**File:** frpkol1 (Project Category Codes)

•**Field:** b1valg

•**Condition:** Specific function keys trigger different subroutine calls based on user actions.

Financial and Transactional Rules

1. Record Update

•**Logic:** When updating a record, the program checks if the record exists and updates the fields accordingly.

•**File:** frpkolr (Project Category Record)

•**Fields:**

•rpktxt (Description)

•rpkeus (User)

•**Condition:** The update occurs if the record is found in the database.

2. Record Deletion

•**Logic:** The program allows users to delete a record after confirming the action.

•**File:** frpkolr (Project Category Record)

•**Condition:** The deletion is executed if the user confirms the action in the delete window.

Special Conditions (Program-Specific)

1. Duplicate Record Check (RP001R)

•**Logic:** Before creating a new record, the program checks if a record with the same key already exists.

•**File:** frpkolr (Project Category Record)

•**Field:** rpkolr_kkod

•**Condition:** If a duplicate is found, a message is displayed to the user, preventing the creation of the new record.

2. Record Copying (RP001R)

•**Logic:** The program allows users to copy an existing record to create a new one.

•**File:** frpkolr (Project Category Record)

•**Fields:** k1kode (Key for Copying), c1kode (New Key)

•**Condition:** The copying process is initiated when the user selects the copy function.

Subprogram Calls Affecting Logic

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

1. dsp_subfile (Display Subfile)

•**Trigger:** Called after subfile records are processed.

•**Logic:** This subroutine handles the display of the subfile and manages user feedback.

•**Impact:** This call ensures that the user interface reflects the current state of the records.

2. crt_subfile (Create Subfile)

•**Trigger:** Called to populate the subfile with records.

•**Logic:** This subroutine reads records from the database and formats them for display in the subfile.

- **Impact:** This is a **critical step** in ensuring that users see up-to-date information.

3. xc2bld (Maintain Record)

- **Trigger:** Called when the user wishes to create or update a record.
- **Logic:** This subroutine manages the logic for maintaining records, including validation and updates.
- **Impact:** Represents the **core functionality** for record management within the program.