

RP001R.MBR

Path: NXCLOUD/rpgsrc/RP001R.MBR **Generated:** 2026-01-08 12:58:56 **Processing Time:** 13053ms

Business Logic for Koder Vedlikehold

This document outlines the business rules that govern the maintenance of codes in the RPG program RP001R. The primary focus is on how codes are managed, displayed, and manipulated within the system.

The core logic for code maintenance is contained within the various subroutines in RP001R. The program handles user interactions through a subfile interface, allowing users to create, update, delete, and view code entries.

Order Status and Header Rules

Koder Vedlikehold: frpkol1, frpkol2, frpkolr, frpkolu, frp001d

1. Record Selection

- **Logic:** The program checks for existing records based on user input and positions the cursor accordingly.
- **File:** frpkol1 (Project Category Codes)
- **Field:** rpkol1_kkod
- **Condition:** The process will not select a record if b2kode is blank or b2text is blank.

2. Subfile Display

- **Logic:** The program displays the subfile with the current records and allows navigation through them.
- **File:** frp001d (Subfile for Code Maintenance)
- **Field:** b1sfl
- **Condition:** The subfile will be cleared and repopulated based on the current state of the database.

Configuration and Authorization Rules

1. User Authorization Check

- **Logic:** The program checks the user's authorization level before allowing certain actions.

• **Files:**

- frpkol1 (Project Category Codes)

- frpkolr (Reference Codes)

• **Fields:**

- l_user (User Identifier)

- w_firm (Firm Identifier)

- **Condition:** The program will only allow updates if the user has the necessary permissions.

2. Firm Type Validation

- **Logic:** The program validates the firm type against predefined categories.

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

- **Field:** w_type

- **Condition:** The process will display an error if the firm type does not match expected values.

Financial and Transactional Rules

1. Record Creation

- **Logic:** When creating a new record, the program checks if the key already exists and updates or creates accordingly.

- **File:** frpkolu (Updated Codes)

• **Fields:**

- rpkkod (Code Identifier)

- rpktxt (Description)

- **Condition:** The record will be created if it does not exist; otherwise, it will prompt for an update.

2. Record Deletion

- **Logic:** The program allows users to delete records after confirmation.

- **File:** frpkolr (Reference Codes)

- **Condition:** The record will be deleted only if the user confirms the action.

Special Conditions (Program-Specific)

1. Subfile Management (RP001R)

- **Logic:** The program manages the subfile's state, including pagination and record navigation.

- **File:** frp001d (Subfile for Code Maintenance)

- **Field:** w_srrn

- **Condition:** The subfile will be updated based on user actions, and the current record number will be tracked.

2. Error Handling (RP001R)

- **Logic:** The program displays error messages based on user input and system state.

- **File:** frp001d (Subfile for Code Maintenance)

- **Fields:** b1sfl (Subfile Record), b2cmd (Command Record)

- **Condition:** Error messages will be displayed if required fields are not filled or if the user attempts invalid operations.

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 to refresh the subfile display after any operation.

- **Logic:** This subroutine clears and repopulates the subfile based on the current state of the database.

- **Impact:** This call acts as a **major logical gateway** for user interactions with the subfile.

2. crt_subfile (Create Subfile)

- **Trigger:** Invoked to populate the subfile with records.

- **Logic:** This subroutine reads records from the database and writes them to the subfile.

- **Impact:** This is a **critical step** for displaying current records to the user.

3. xc2bld (Maintenance Program Call)

- Trigger:** Called when creating or updating a record.
- Logic:** This subroutine handles the logic for maintaining code records, including validation and updates.
- Impact:** Represents the handoff to the next major business function, ensuring data integrity during updates.