VA110R.MBR

Path: NXCLOUD/rpgsrc/VA110R.MBR **Generated**: 2026-01-09 10:13:45 **Processing Time**: 11882ms

Business Logic for VA110R

This document outlines the business rules that govern the maintenance of units in the ASOFAK system, based on an analysis of the RPG program VA110R. The primary focus is on the handling of subfiles, user interactions, and data management for unit maintenance.

The core logic for unit maintenance is contained within the various subroutines in VA110R. The program processes user inputs, manages subfile data, and performs CRUD operations on unit records.

Order Status and Header Rules

VA110R: venhl1, venhl2, venhlr, venhlu

1. Record Selection

- **Logic:** The program will select records based on user input for either unit number or description.
- **File:** venhl1 (Unit header file)
- **Field:** venhl1_eenh
- **Condition:** The process will not select a record if b2eenh is blank.

2. Subfile Display

- **Logic:** The program displays the subfile based on the current record number and page size.
- **File:** b1sfl (Subfile for displaying unit records)
- **Field:** b1eenh
- **Condition:** The subfile will be cleared and repopulated when the user navigates or performs actions.

Configuration and Authorization Rules

1. User Authorization Check

- **Logic:** The program checks if the user has the necessary permissions to perform actions on the units.
- **Files:**
- venhlr (Unit records)
- venhlu (Unit update records)
- **Fields:**
- venhlr_eenh (Unit number)
- venhlu_eenh (Unit number)
- **Condition:** The user must be authorized to update or delete records based on their role.

2. Input Validation

- **Logic:** The program validates user input before processing.
- **File:** c1bld (Input screen for creating new units)
- **Field:** c1eenh
- **Condition:** The process will prompt an error if c1eenh is blank during creation.

Financial and Transactional Rules

1. Record Update Logic

•**Logic:** When a record is updated, the program captures the user and timestamp for auditing.

•**File:** venhlu (Unit update records)

•**Fields:**

•vaeeus (User who made the update)

•vaeoda (Date of update)

•**Condition:** The update will only occur if the record exists in the database.

2. Record Deletion Logic

•**Logic:** The program allows for the deletion of unit records after user confirmation.

•**File:** venhlu (Unit update records)

•**Condition:** The deletion will only proceed if the user confirms the action in the delete window.

Special Conditions (Program-Specific)

1. Subfile Initialization (VA110R)

•**Logic:** The program initializes the subfile and sets up keys for database access.

•**File:** venhl1 (Unit header file)

•**Field:** venhl1_eenh

•**Condition:** The subfile is populated only if the firm identifier is valid.

2. Error Handling (VA110R)

•**Logic:** The program provides feedback to the user when an error occurs.

•**File:** c1msg (Error message screen)

•**Fields:** b_anul (Cancel flag)

•**Condition:** The error message is displayed if a duplicate entry is detected during creation.

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 user actions to refresh the subfile display.

•**Logic:** This subroutine manages the visibility and contents of the subfile.

•**Impact:** This call ensures the user sees the most current data, acting as a **major logical gateway** for user interactions.

2. crt_subfile (Create Subfile)

•**Trigger:** Invoked when initializing or refreshing the subfile.

•**Logic:** This subroutine populates the subfile with records based on the current context.

•**Impact:** This is a **destructive filtering step** that clears previous data and repopulates it based on the latest user input.

3. xc2bld (Maintenance Program Call)

•**Trigger:** Called when the user wants to create or update a unit record.

•**Logic:** This subroutine handles the logic for maintaining unit records.

- **Impact:** Represents the handoff to the next major business function, ensuring that all updates are processed correctly.