

VA110R.MBR

Path: NXCLOUD/rpgsrc/VA110R.MBR **Generated:** 2026-01-08 13:33:10 **Processing Time:** 11321ms

Business Logic for VA110R

This document outlines the business rules that govern the maintenance of units within the ASOFAK system, based on an analysis of the RPG program VA110R. The primary focus is on how the program manages subfile operations for unit entries, including creation, updating, and deletion.

The core logic for processing unit maintenance is contained within the various subroutines such as forny, posisjoner, and subfile in VA110R. The program handles user interactions through function keys and manages data display via subfiles.

Order Status and Header Rules

VA110R: venhl1, venhl2, venhlr, venhlu

1. Unit Entry Display

• **Logic:** Displays unit entries in a subfile format, allowing users to view and navigate through records.

• **File:** venhl1 (Unit header file)

• **Field:** venhl1_eenh

• **Condition:** The program will not display records if the unit entry field is blank.

2. Unit Description Display

• **Logic:** Displays descriptions associated with unit entries in the subfile.

• **File:** venhl2 (Unit description file)

• **Field:** venhl2_etxt

• **Condition:** The program will not display records if the description field is blank.

Configuration and Authorization Rules

1. User Authorization Check

• **Logic:** Validates user permissions before allowing modifications to unit entries.

• **Files:**

• venhlr (Unit reference file)

• venhlu (Unit update file)

• **Fields:**

• venhlr_eenh (from venhlr)

• venhlu_eenh (from venhlu)

• **Condition:** The program checks if the user has the necessary permissions before allowing updates.

2. Firm Validation

• **Logic:** Ensures that the firm associated with the unit entry matches the user's firm.

• **File:** venhl1 (Unit header file)

• **Field:** w_firm

- **Condition:** The process will not proceed if the firm does not match the user's firm.

Financial and Transactional Rules

1. Unit Creation and Update

- **Logic:** Handles the creation and updating of unit entries, including timestamping and user tracking.

- **File:** venhlu (Unit update file)

- **Fields:**

- vaetxt (Unit text description)

- vaeeda (Creation date)

- **Condition:** The program will only create or update records if the necessary fields are populated.

2. Unit Deletion

- **Logic:** Manages the deletion of unit entries from the database.

- **File:** venhlu (Unit update file)

- **Condition:** The program will prompt for confirmation before deletion and will only delete if the entry exists.

Special Conditions (Program-Specific)

1. Subfile Management (VA110R)

- **Logic:** Manages the display and navigation of records in a subfile format, handling paging and record selection.

- **File:** b1sfl (Subfile for unit entries)

- **Field:** w_srrn

- **Condition:** The program will not display records if there are no entries in the subfile. Note: The subfile management is crucial for user interaction.

2. Error Handling (VA110R)

- **Logic:** Displays error messages when invalid operations are attempted, such as duplicate entries.

- **File:** c1msg (Message display file)

- **Fields:** c1eenh (Entry key), c1etxt (Error text)

- **Condition:** The program will show an error message if an attempt is made to create a duplicate entry.

Subprogram Calls Affecting Logic

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

1. xc2bld (Unit Maintenance Program)

- **Trigger:** Called when creating or updating a unit entry.

- **Logic:** Validates input and updates the unit entry in the database.

- **Impact:** This call ensures that all necessary validations are performed before any database changes are made.

2. xc1bld (Unit Creation Program)

- Trigger:** Called when a new unit entry is being created.
- Logic:** Initializes the entry fields and prepares the screen for user input.
- Impact:** This is a critical step for ensuring that new entries are created correctly.

3. xd1win (Delete Confirmation Window)

- Trigger:** Called when a delete operation is initiated.
- Logic:** Displays a confirmation window to the user before proceeding with deletion.
- Impact:** This represents a safeguard against accidental deletions, enhancing user experience and data integrity.