

# VA110R.MBR

**Path:** NXCLOUD/rpgsrc/VA110R.MBR **Generated:** 2026-01-08 12:35:33 **Processing Time:** 11753ms

## 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 the handling of subfiles and the user interface for maintaining unit records.

The core logic for the maintenance of units is contained within various subroutines in VA110R. The program processes user inputs, manages subfile data, and interacts with the database to create, update, and delete unit records.

### Order Status and Header Rules

VA110R: venhl1, venhl2, venhlr, venhlu

#### 1. Subfile Display Logic

- **Logic:** The program displays a subfile containing unit records based on user interaction and database content.
- **File:** venhl1 (Unit header file)
- **Field:** venhl1\_eenh
- **Condition:** The subfile will be populated if the user has not provided any specific search criteria (i.e., b2eenh or b2etxt is not blank).

#### 2. Record Navigation

- **Logic:** The program allows navigation through the records in the subfile, enabling users to roll up and down through the records.
- **File:** venhl1 (Unit header file)
- **Field:** w\_srrn
- **Condition:** Navigation is enabled when the subfile contains records (w\_srrn is not equal to 0).

### Configuration and Authorization Rules

#### 1. User Authorization for Operations

- **Logic:** The program checks user permissions before allowing operations such as create, update, or delete.

##### • **Files:**

- venhl1 (Unit header file)
- venhlr (Unit reference file)

##### • **Fields:**

- l\_user (Current user)
- w\_firm (Current firm)

- **Condition:** Operations are allowed only if the user has the appropriate permissions set in the system.

#### 2. Record Existence Check Before Update

- **Logic:** The program verifies if a record exists before allowing updates or deletions.

- **File:** venhlr (Unit reference file)
  - **Field:** venhlr\_eenh
  - **Condition:** The program checks for the existence of the record using a chain operation before proceeding with any update.
- 

## Financial and Transactional Rules

### 1. Update Logic for Unit Records

- **Logic:** When a unit record is updated, the program captures the current user and timestamps the update.
- **File:** venhlru (Unit update file)
- **Fields:**
  - vaeeus (User updating the record)
  - vaeoda (Update timestamp)
- **Condition:** The update occurs only if the record is found in the database.

### 2. Delete Logic for Unit Records

- **Logic:** The program allows deletion of unit records after confirming the user's intention through a confirmation screen.
  - **File:** venhlr (Unit reference file)
  - **Condition:** The deletion process is initiated only after the user confirms the action in the delete confirmation window.
- 

## Special Conditions (Program-Specific)

### 1. Subfile Initialization (VA110R)

- **Logic:** The program initializes the subfile by setting up keys and preparing the display for user interaction.
- **File:** venhl1 (Unit header file)
- **Field:** venhl1\_key
- **Condition:** The subfile is initialized with blank values when the program starts.

### 2. Error Handling for Existing Records (VA110R)

- **Logic:** If a user tries to create a record that already exists, the program displays an error message.
  - **File:** venhlr (Unit reference file)
  - **Fields:** venhlr\_eenh (Existing unit key)
  - **Condition:** The program checks for existing keys before allowing the creation of a new record.
- 

## 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 display the contents of the subfile after user interaction.
- **Logic:** This subroutine manages the display of records in the subfile based on the current record number and user inputs.

- Impact:** This call acts as a **major logical gateway** for user navigation through unit records.

## **2. crt\_subfile (Create Subfile)**

- Trigger:** Called to populate the subfile with records from the database.

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

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

## **3. xc2bld (Maintenance Program Call)**

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

- Logic:** This subroutine handles the logic for maintaining unit records, including validation and database updates.

- Impact:** Represents the handoff to the next major business function of maintaining unit records.