

# VP110R.MBR

**Path:** NXCLOUD/rpgsrc/VP110R.MBR **Generated:** 2026-01-08 13:33:18 **Processing Time:** 13136ms

## Business Logic for Vare-pris Maintenance

This document outlines the business rules that govern the maintenance of item prices in the ERP system, based on an analysis of the RPG program VP110R. The primary focus is on the logic related to price management, including the addition, deletion, and display of item prices.

The core logic for price maintenance is contained within the initiating av program subroutine in VP110R. The program processes item prices associated with various vendors and includes functionality for handling subfiles that display these prices.

## Order Status and Header Rules

Vare-pris Maintenance: vvprlr, vvprl1, vvprlu, vvarl1, vvenl1, vltp1

### 1. Price Deletion Check

- Logic:** The program checks if an item price can be deleted based on existing orders. If there are orders associated with the item, deletion is not permitted.

- File:** vvprlr (Item Price File)

- Field:** b\_best

- Condition:** "The process will not allow deletion if b\_best is set to \*on, indicating that there are existing orders for the item."

### 2. Price Group Validation

- Logic:** The program verifies if a price group exists for the item before allowing any operations related to that price group.

- File:** vvprlr (Item Price File)

- Field:** b\_prgr

- Condition:** "The process will not proceed if b\_prgr is set to \*off, indicating that the price group is not valid."

## Configuration and Authorization Rules

### 1. Vendor Price Group Check

- Logic:** When deleting a price, the program checks if the price is associated with a vendor. If the vendor's last price is deleted, the vendor is removed from the inventory.

- Files:**

- vvprlr (Item Price File)

- vvenl1 (Vendor File)

- Fields:**

- vvprlr\_enhe (from vvprlr)

- vvenlu\_enhe (from vvenl1)

- Condition:** "The deletion will only proceed if there are no remaining prices for the vendor."

### 2. Subfile Control Logic

- Logic:** The program manages the display of item prices in a subfile, ensuring that the user interface reflects the current state of prices.

- File:** b1sfl (Subfile)
  - Field:** w\_srrn
  - Condition:** "The subfile will not display records if w\_srrn equals 0, indicating no records are available."
- 

## Financial and Transactional Rules

### 1. Price Calculation Logic

- Logic:** The program calculates the coverage degree for item prices based on the purchase price and sales price.
- File:** vvprlr (Item Price File)
- Fields:**
  - b1sapr (Sales Price)
  - b1kopr (Purchase Price)
- Condition:** "If b1sapr is not zero, the program calculates the coverage degree; otherwise, it retrieves the default coverage degree."

### 2. Price Update Logic

- Logic:** When updating prices, the program checks if the new price is valid and updates the relevant records accordingly.
  - File:** vvprlu (Updated Item Price File)
  - Condition:** "The update will only occur if the new price passes validation checks against existing prices."
- 

## Special Conditions (Program-Specific)

### 1. Price Group Blank Check (VP110R)

- Logic:** The program checks if a blank price group exists for an item before allowing deletion.
- File:** vvprlr (Item Price File)
- Field:** b\_pgbl
- Condition:** "The deletion will only proceed if b\_pgbl is set to \*on, indicating that a blank price group is valid."

### 2. Order Existence Check (VP110R)

- Logic:** The program checks if there are any orders associated with the item before allowing deletion.
  - File:** lodtlv (Order Details File)
  - Fields:** lldldor, ldenh1
  - Condition:** "The deletion will not proceed if b\_best is set to \*on, indicating that there are existing orders for the item."
- 

## Subprogram Calls Affecting Logic

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

### 1. CO402R (Price Group Check)

- Trigger:** Called during program initialization to check for valid price groups.

- Logic:** This program checks if the price group is valid and sets the u\_701 flag accordingly.

- Impact:** "This call acts as a **validation step** for price groups before any price operations can occur."

## 2. VL711R (Vendor Price Check)

- Trigger:** Called conditionally based on the u\_701 flag.

- Logic:** This program checks vendor prices and updates the necessary fields.

- Impact:** "This is a **critical check** to ensure that vendor prices are valid before proceeding with price maintenance."

## 3. xh1win (Window Handling)

- Trigger:** Called for handling user interface interactions.

- Logic:** This program manages the display of the price maintenance window.

- Impact:** "This represents the **user interface management** for the price maintenance process, ensuring a smooth user experience."