

# JX100R.MBR

**Path:** NXCLOUD/rpgsrc/JX100R.MBR **Generated:** 2026-01-09 10:13:44 **Processing Time:** 13125ms

## 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 JX100R. The primary focus is on how item prices are managed, updated, and validated within the system.

The core logic for price maintenance is contained within the various subroutines in JX100R. The program processes records related to item pricing, including creation, updates, deletions, and validations against business rules.

## Order Status and Header Rules

**Vare-pris Maintenance:** fjvprlr, fjvprl1, fjvprlu, fjvarl1, fjlevl1, fjvlelr, favall1

### 1. Check for Valid From Date

- Logic:** The program checks whether a valid "from" date is filled out when creating or updating item prices.
- File:** fjvprlr (Item Price Records)
- Field:** jvprlr\_gdat
- Condition:** "The process will not proceed if the jvprlr\_gdat is not filled out."

### 2. Update Price with Old Price

- Logic:** When updating an item price, the old price is retained for reference.
- File:** fjvprlu (Item Price Update Records)
- Field:** jvprlu\_rsts
- Condition:** "The old price will be updated if the jvprlu\_rsts is set to 1."

## Configuration and Authorization Rules

### 1. Field Write Protection Based on User Roles

- Logic:** Certain fields are protected from editing based on user roles and configurations.
- Files:**
  - favall1 (Field Authorization)
- Fields:**
  - avall1\_apgm (Program Name)
- Condition:** "Fields will be write-protected if the corresponding authorization record indicates so."

### 2. Display Expired Prices

- Logic:** The program allows expired prices to be displayed when accessing the price maintenance screen for the first time.
- File:** fjvprlr (Item Price Records)
- Field:** jvprlr\_rsts
- Condition:** "Expired prices will be shown if the jvprlr\_rsts is set to 0."

## Financial and Transactional Rules

## 1. Price Update Logic

- Logic:** When updating prices, the program checks if the price already exists and updates it accordingly.
- File:** fjvprlu (Item Price Update Records)
- Fields:**
  - jvprlu\_idor (Supplier ID)
  - jvprlu\_gdat (Price Date)
- Condition:** "The price will be updated if an existing record is found for the supplier and date."

## 2. Delete Price Record

- Logic:** The program allows for the deletion of price records after confirmation.
  - File:** fjvprlu (Item Price Update Records)
  - Condition:** "A record will be deleted if the user confirms the deletion action."
- 

# Special Conditions (Program-Specific)

## 1. Display Message for Existing Record (JX100R)

- Logic:** If a user tries to create a price record that already exists, a message is displayed.
- File:** fjvprlr (Item Price Records)
- Field:** jvprlr\_idor
- Condition:** "A message will be shown if the jvprlr\_idor already exists in the database."

## 2. Check for Valid Input on Price Creation (JX100R)

- Logic:** The program validates that all required fields are filled before allowing the creation of a new price record.
  - File:** fjvprlr (Item Price Records)
  - Fields:** jvprlr\_idor, jvprlr\_gdat
  - Condition:** "The creation will be blocked if jvprlr\_idor or jvprlr\_gdat is not provided."
- 

# Subprogram Calls Affecting Logic

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

## 1. JK101R (Logistics Item Update)

- Trigger:** Called during price updates to ensure logistics data is synchronized.
- Logic:** Updates logistics-related information for the item.
- Impact:** "This call ensures that logistics data remains consistent with the updated pricing."

## 2. JV790R (Search Text Update)

- Trigger:** Called after a price update to refresh search text data.
- Logic:** Updates the search text associated with the item.
- Impact:** "This is a crucial step to ensure that search functionalities reflect the latest pricing information."

## 3. JL500R (Supplier Lookup)

- Trigger:** Called during various input validations to fetch supplier details.
- Logic:** Retrieves supplier information based on the provided supplier ID.

•**Impact:** "This call is essential for ensuring that all supplier-related data is accurate and up-to-date."