

# VT100R.MBR

**Path:** NXCLOUD/rpgsrc/VT100R.MBR **Generated:** 2026-01-08 12:59:21 **Processing Time:** 14438ms

## Business Logic for VT100R

This document outlines the business rules that govern the process of managing offer prices and maintenance within the RPG program VT100R. The primary focus is on how offer prices are handled, validated, and updated in the system.

The core logic for managing offer prices is contained within the forny subroutine in VT100R. The program processes various records related to offer prices, including validation and updating of these prices based on user input and existing data.

### Order Status and Header Rules

**VT100R:** vtillr, vti2lr, vtillu, vti2lu, vtil1, vtil2, vtil3, vtil4, vtil5, vtil6, vtil8, vtil9, vvarl1, vtpl1, fra30l1, vvenl1, frlevl1, fvogrl1, fvhgrl1, fvugrl1

#### 1. Check for Valid Item

- **Logic:** The process will not proceed if the item does not exist in the database.
- **File:** vtillr (Offer Prices)
- **Field:** vtillr\_vare
- **Condition:** "The process will not select a record if vtillr\_vare is not found."

#### 2. Price Group Inclusion

- **Logic:** The price group must be included when maintaining offer prices.
- **File:** vtillr (Offer Prices)
- **Field:** vtillr\_prgr
- **Condition:** "The process will not select a record if vtillr\_prgr is not specified."

### Configuration and Authorization Rules

#### 1. User Authorization Check

- **Logic:** The program checks if the user has the necessary permissions to modify offer prices.
- **Files:**
  - vtillr (Offer Prices)
  - vvenl1 (Vendor Information)
- **Fields:**
  - vtillr\_firm (from vtillr)
  - vvenl1\_enhe (from vvenl1)
- **Condition:** "The process will not allow updates if the user is not authorized."

#### 2. Date Validation

- **Logic:** The program validates that the start and end dates for the offer are correct.
- **File:** vtillr (Offer Prices)
- **Field:** vtillr\_fdat, vtillr\_tdat
- **Condition:** "The process will not proceed if vtillr\_fdat is greater than vtillr\_tdat."

# Financial and Transactional Rules

## 1. Price Calculation

•**Logic:** The program calculates the offer price based on the input values and existing vendor prices.

•**File:** vtillr (Offer Prices)

•**Fields:**

•vtillr\_pgr (Price Group)

•vtillr\_lety (Delivery Type)

•**Condition:** "The calculation will not occur if the price group or delivery type is missing."

## 2. Update Offer Price

•**Logic:** The program updates the offer price in the database based on user input.

•**File:** vtillr (Offer Prices)

•**Condition:** "The update will only occur if all validation checks pass."

# Special Conditions (Program-Specific)

## 1. Price Change Notification (VT100R)

•**Logic:** The program notifies the user if the price has changed for a specific item.

•**File:** vtillr (Offer Prices)

•**Field:** s\_tip1, s\_tip2, s\_tip3

•**Condition:** "The notification will trigger if the current price differs from the previously stored price."

## 2. Subfile Management (VT100R)

•**Logic:** The program manages subfile operations for displaying offer prices.

•**File:** b1sfl (Subfile)

•**Fields:** b1kamp, b1enh1, b1tip1

•**Condition:** "The subfile will be updated based on user selections and actions."

# Subprogram Calls Affecting Logic

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

## 1. VP715R (Price Calculation Program)

•**Trigger:** Called during the price calculation process.

•**Logic:** This program retrieves and calculates the necessary price data based on the provided parameters.

•**Impact:** "This call acts as a **critical component** in determining the final offer price."

## 2. VG510R (Group Information Retrieval)

•**Trigger:** Called when validating group information.

•**Logic:** This program checks if the provided group exists and retrieves its details.

•**Impact:** "This ensures that all group-related data is valid before proceeding."

## 3. VV500R (Item Information Retrieval)

•**Trigger:** Called when validating item numbers.

- Logic:** This program checks if the item exists in the inventory.

- Impact:** "This represents a **key validation step** to ensure the item is valid before processing."