

# VA551R.MBR

**Type:** Single File Analysis **Path:** NXCLOUD/rpgsrc/VA551R.MBR **Generated:** 2026-01-29

11:53:59 **Processing Time:** 18862ms

-

## # Business Logic for Order Type Inquiry

This document outlines the business rules that govern the inquiry and processing of order types, based on an analysis of the RPG program `VA551R`. The primary focus is on the conditions that will **block** an order type from being displayed or processed in the subfile.

The core logic for managing subfile records and blocking conditions is contained within the `crt\_subfile` subroutine in `VA551R`. An order type must pass all the following checks to be displayed in the subfile.

## ## Order Status and Header Rules

### 1. \*\*Order Type Filtering\*\*

\* **Logic:** Only order types that are valid for invoicing (accumulator type 2) are included in the subfile.

\* **File:** `VOTYL1` (Order Type File)

\* **Field:** `VAOAKK`

\* **Condition:** The subfile will not display an order type if `VAOAKK` is not equal to `2`. This ensures only order types eligible for invoicing are shown.

### 2. \*\*Firm Validation\*\*

\* **Logic:** The firm associated with the order type must match the firm parameter provided to the program.

\* **File:** `VOTYL1` (Order Type File)

\* **Field:** `VAOFIR`

\* **Condition:** The subfile will not display an order type if `VAOFIR` does not match the value of `W\_FIRM`. This ensures firm-specific filtering.

## ## Configuration and Authorization Rules

### 3. \*\*Wildcard Filtering\*\*

\* **Logic:** Order types starting with `\*` are treated as special cases and may be excluded based on specific rules.

\* **File:** `VOTYL1` (Order Type File)

\* **Field:** `VAOTYP`

\* **Condition:** The subfile will not display an order type if the first character of `VAOTYP` is `\*` and the second character does not match the second character of the `B2OTYP` field. This ensures wildcard filtering logic is applied correctly.

## ## Financial and Transactional Rules

### 4. \*\*Subfile Record Limits\*\*

\* **Logic:** The subfile has a maximum page size and record limits that

control how many order types can be displayed at once.

- \* \*\*File:\*\* `VA551D` (Subfile Display File)

- \* \*\*Field:\*\* `W\_SRRN`, `W\_SPGE`

- \* \*\*Condition:\*\* The subfile will stop adding records once the relative record number (`W\_SRRN`) reaches the page size (`W\_SPGE`). This prevents overflow in the subfile.

## Special Conditions (Program-Specific)

#### 5. \*\*EOF Handling\*\*

- \* \*\*Logic:\*\* If the end of file (EOF) is reached during subfile creation or navigation, no further records are processed.

- \* \*\*File:\*\* `VOTYL1` or `VOTYL2` (Order Type Files)

- \* \*\*Field:\*\* N/A (EOF condition)

- \* \*\*Condition:\*\* The subfile processing will terminate if `%EOF` is true during record reads. This ensures proper handling of file boundaries.

#### 6. \*\*Selection Logic\*\*

- \* \*\*Logic:\*\* Users can select an order type from the subfile for further processing.

- \* \*\*File:\*\* `VA551D` (Subfile Display File)

- \* \*\*Field:\*\* `B1VALG`

- \* \*\*Condition:\*\* The program will proceed to finalize the selection if `B1VALG` is equal to `1`. This ensures only selected records are processed.

## Subprogram Calls Affecting Logic

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

#### 1. \*\*`CLR\_SUBFILE` (Clear Subfile)\*\*

- \* \*\*Trigger:\*\* Called before creating or refreshing the subfile.

- \* \*\*Logic:\*\* Clears all subfile records and resets subfile-related variables (`W\_SRRN`, `W\_SPGE`, etc.).

- \* \*\*Impact:\*\* Ensures the subfile is empty before new records are added.

#### 2. \*\*`CRT\_SUBFILE` (Create Subfile)\*\*

- \* \*\*Trigger:\*\* Called during initial program setup and when refreshing the subfile.

- \* \*\*Logic:\*\* Reads records from `VOTYL1` or `VOTYL2` based on positioning keys and adds them to the subfile.

- \* \*\*Impact:\*\* Populates the subfile with valid order types based on filtering rules.

#### 3. \*\*`BCK\_SUBFILE` (Back Subfile)\*\*

- \* \*\*Trigger:\*\* Called when navigating backward in the subfile.

- \* \*\*Logic:\*\* Reads records in reverse order and repopulates the subfile for the previous page.

- \* \*\*Impact:\*\* Allows navigation to previous pages while maintaining

filtering rules.

4. **\*\*`POSIJONER` (Position Subfile)\*\***

\* **\*\*Trigger:\*\*** Called when specific positioning criteria are provided (e.g., order type or description).

\* **\*\*Logic:\*\*** Sets positioning keys and adjusts subfile records accordingly.

\* **\*\*Impact:\*\*** Ensures the subfile reflects the requested position criteria.

5. **\*\*`DSP\_SUBFILE` (Display Subfile)\*\***

\* **\*\*Trigger:\*\*** Called to display the subfile on the screen.

\* **\*\*Logic:\*\*** Handles screen formatting and user interaction with the subfile.

\* **\*\*Impact:\*\*** Provides the user interface for subfile navigation and selection.