

# NO100R.MBR

**Path:** NXCLOUD/rpgsrc/NO100R.MBR **Generated:** 2026-01-08 12:35:06 **Processing Time:** 15127ms

## Business Logic for External Order Header Maintenance

This document outlines the business rules that govern the maintenance of external order headers, based on an analysis of the RPG program NO100R. The primary focus is on the logic that handles the processing of external orders, including validation, updates, and interactions with other components of the system.

The core logic for maintaining external order headers is contained within the various subroutines in NO100R. The program processes records from multiple files, performs validations, updates records, and interacts with other programs to ensure the integrity and accuracy of the order data.

### Order Status and Header Rules: NOHEL2, NOHEL3, NOHEL4, NOHEL5, NOHEL6

#### 1. Check for Existing Orders

- **Logic:** The program checks if an external order already exists in the system before allowing updates or transfers.

- **File:** NOHEL2 (External Order Header)

- **Field:** NOHEL2\_NREF

- **Condition:** The order will not be processed if NOHEL2\_NREF is not blank or if the order is already marked as transferred.

#### 2. Validate Customer Information

- **Logic:** The program ensures that customer information is valid and complete before proceeding with order processing.

- **File:** NOHEL3 (Customer Information)

- **Field:** NOHEL3\_KNAV

- **Condition:** The process will not proceed if NOHEL3\_KNAV is blank.

#### 3. Check for Locked Orders

- **Logic:** The program checks if the order is locked in another process before allowing modifications.

- **File:** NOHEL1 (Locked Orders)

- **Field:** NOHEL1\_TELL

- **Condition:** The order will not be modified if it is found to be locked.

### Configuration and Authorization Rules: NOHEL1, NOHEL2, NOHEL3, NOHEL4, NOHEL5

#### 1. Authorization for Order Creation

- **Logic:** The program checks user permissions before allowing the creation of new orders.

- **Files:**

- NOHEL1 (User Permissions)

- NOHEL2 (External Order Header)

**• Fields:**

- NOHEL1\_USER (from NOHEL1)
- NOHEL2\_NREF (from NOHEL2)

**• Condition:** The order will not be created if the user does not have the necessary permissions.

## 2. Check for Valid Delivery Methods

**• Logic:** The program validates that the selected delivery method is valid and available.

**• File:** NOHEL4 (Delivery Methods)

**• Field:** NOHEL4\_PNAV

**• Condition:** The process will not proceed if NOHEL4\_PNAV is invalid.

# Financial and Transactional Rules: NODTL1, NODTL2

## 1. Update Order Lines with Delivery Dates

**• Logic:** The program updates order lines with delivery dates based on the main order header.

**• File:** NODTL1 (Order Lines)

**• Fields:**

- NODTL1\_LDAT (Delivery Date)

- NODTL1\_VARE (Item)

**• Condition:** The delivery date will be updated if it is not blank in the order header.

## 2. Check for Inventory Availability

**• Logic:** The program checks if the items in the order are available in inventory before processing the order.

**• File:** NODTL1 (Order Lines)

**• Condition:** The order will not be processed if any item is not available in inventory.

# Special Conditions (Program-Specific)

## 1. Handle EDI Orders

**• Logic:** The program processes EDI orders differently, ensuring they meet specific criteria before being accepted.

**• File:** NOHEL2 (External Order Header)

**• Field:** NOHEL2\_FSYS

**• Condition:** The order will not be processed if it is an EDI order and does not meet the required standards.

## 2. Temporary Status for Orders in Transition

**• Logic:** If an order is in the process of being transferred, it is marked with a temporary status to prevent further modifications.

**• File:** NOHEL2 (External Order Header)

**• Field:** NOHEL2\_TELL

**• Condition:** The order will be marked with a temporary status if it is currently being transferred.

# Subprogram Calls Affecting Logic

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

## **1. NO110R (Status Codes)**

- Trigger:** This program is called when the user requests status codes.
- Logic:** It retrieves and displays the current status codes for the order.
- Impact:** This call acts as a **major logical gateway** for understanding the order's current state.

## **2. NO150R (Customer Order Retrieval)**

- Trigger:** This program is called when the user requests to retrieve a customer order.
- Logic:** It fetches the details of the specified customer order.
- Impact:** This is a **critical step** for ensuring that the correct order details are processed.

## **3. FA730R (Job Status Update)**

- Trigger:** This program is called to check if a job is currently running.
- Logic:** It updates the job status to prevent conflicts during order processing.
- Impact:** This ensures that **overlapping processes** do not interfere with order handling.

## **4. ND105R (Order Line Processing)**

- Trigger:** This program is called to handle individual order lines.
- Logic:** It processes the details of each line item in the order.
- Impact:** This represents the **detailed handling** of order line items, ensuring accuracy in processing.

## **5. NO120R (Transfer Orders)**

- Trigger:** This program is called to initiate the transfer of orders to the next stage.
- Logic:** It handles the logic for transferring orders, including validation checks.
- Impact:** This is a **key step** in the order processing workflow, determining whether an order can proceed to the next phase.

Use clear, business-friendly language. Fill in all sections with details from the code. Output only the Markdown.