NO100R.MBR

Path: NXCLOUD/rpgsrc/NO100R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 14656ms

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 processing of external orders and their associated logic.

The core logic for maintaining external order headers is contained within the various subroutines in NO100R. The program handles the reading, updating, and deletion of order headers and their associated lines, ensuring that all necessary validations and business rules are applied.

Order Status and Header Rules

External Order Maintenance: NOHEL2, NOHEL3, NOHEL4, NOHEL5, NOHEL6, NOHELR

1. Check for Existing Orders

•**Logic:** The program checks if an order already exists in the database before creating a new one.

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

•**Field:** NOHEL2_NREF

•**Condition:** The process will not create a new order if NOHEL2_NREF is not blank.

2. Update Order Header Information

•**Logic:** Updates the order header with the latest information when an order is modified.

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

•**Field:** NOHELUR_TELL, NOHELUR_FSYS

•**Condition:** The header is updated only if the order exists in the database.

Configuration and Authorization Rules

1. EDI Processing Check

•**Logic:** Determines if the order is being processed as an EDI order and sets flags accordingly.

•**Files:**

•NOHELUR (External Order Header)

•NOHEL1 (External Order Lines)

•**Fields:**

•NOHELUR_FSYS (from NOHELUR)

•NOHEL1_FSYS (from NOHEL1)

•**Condition:** The process checks if the system type is EDI, SMK, or EHF to apply specific logic.

2. Locking Mechanism for Orders

•**Logic:** Prevents modifications to orders that are currently locked.

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

•**Field:** NOHELUR_TELL

•**Condition:** The order cannot be modified if it is locked (checked against NONUMM).

Financial and Transactional Rules

1. Transfer of Inventory Items

•**Logic:** Handles the transfer of inventory items associated with an order.

•**File:** NODTLR (Order Lines)

•**Fields:**

•NODTLR_FSYS (System Type)

•NODTLR_TELL (Order Number)

•**Condition:** The transfer will not proceed if the order is already marked as transferred.

2. Error Handling for Transfers

•**Logic:** Displays error messages if the transfer conditions are not met.

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

•**Condition:** If the order status is not valid (greater than 0), an error message is generated.

Special Conditions (Program-Specific)

1. EDI Order Lock Check (NO120R)

•**Logic:** Checks if the order is locked in the EDI processing program before allowing modifications.

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

•**Field:** NOHELUR_TELL

•**Condition:** The order cannot be modified if it is locked in the EDI processing program.

2. Creating New Customers and Projects (FA930R, FL104R, FL105R)

•**Logic:** Allows the creation of new customers and projects if they do not already exist.

•**Files:**

•FA930R (Customer Creation)

•FL104R (Project Creation)

•FL105R (Project Creation)

•**Condition:** The program checks for existing customers or projects before creation.

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 Handling)

•**Trigger:** Called when the F13 function key is pressed.

•**Logic:** Retrieves and displays status codes associated with the order.

•**Impact:** This call acts as a **status verification step** for the order.

2. NO150R (Customer Order Retrieval)

•**Trigger:** Called when the F14 function key is pressed.

•**Logic:** Retrieves customer orders based on the current context.

•**Impact:** This is a **critical step** for ensuring that customer orders are correctly linked to external orders.

3. FA730R (Job Status Check)

•**Trigger:** Called to check if a transfer job is currently running.

•**Logic:** Updates job status flags based on the current job state.

•**Impact:** This affects the overall process by determining if further actions can be taken.

4. ND105R (Line Status Retrieval)

•**Trigger:** Called to retrieve the status of order lines.

•**Logic:** Checks the status of each line item associated with the order.

•**Impact:** This is essential for ensuring that all line items are in a valid state before processing.

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