

LO101R.MBR

Path: NXKORR/rpgsrc/LO101R.MBR **Generated:** 2026-01-08 15:53:46 **Processing Time:** 17947ms

Business Logic for LO101R

This document outlines the business rules that govern the registration of orders and purchases, based on an analysis of the RPG program LO101R. The primary focus is on the logic surrounding the validation and updating of order header information.

The core logic for processing order registration is contained within the *inzsr subroutine in LO101R. This subroutine initializes the program, sets up keys for various data structures, and handles the retrieval and validation of order header details.

Order Status and Header Rules

LO101R: LOHEL1, LOHELU, SIHELA, RLTRL7, VVARL1, VLTYL1, LODTLU, VMVAL1, RP1PL1

1. Check for Existing Order Header

- **Logic:** The program checks if an order header exists in the LOHEL1 file. If found, it retrieves various fields such as order type, department, and supplier.
- **File:** LOHEL1 (Order Header Register)
- **Field:** LOMKOD
- **Condition:** The order header is only processed if it is found in the file.

2. Validate Department Code

- **Logic:** The program checks if the department code exists in the code register. If not found, it sets an error indicator.
- **File:** KODE-REGISTER
- **Field:** C2AVDE
- **Condition:** The process will not proceed if the department code does not exist.

Configuration and Authorization Rules

1. Validate Purchaser Code

- **Logic:** The program checks if the purchaser code exists in the code register. If not found, it sets an error indicator.
- **Files:**
- KODE-REGISTER (Purchaser Code Register)
- **Fields:**
- C2INNK (Purchaser Code)
- **Condition:** The process will not proceed if the purchaser code does not exist.

2. Validate Warehouse Code

- **Logic:** The program checks if the warehouse code exists in the code register. If not found, it sets an error indicator.
- **File:** KODE-REGISTER (Warehouse Code Register)
- **Field:** C2LAGE
- **Condition:** The process will not proceed if the warehouse code does not exist.

Financial and Transactional Rules

1. Validate Payment Terms

• **Logic:** The program checks if the payment terms exist in the code register. If not found, it sets an error indicator.

• **File:** KODE-REGISTER (Payment Terms Register)

• **Field:** C2BETB

• **Condition:** The process will not proceed if the payment terms do not exist.

2. Check Invoice Date Validity

• **Logic:** The program checks if the invoice date is valid. If it is not, an error message is displayed.

• **File:** LOHEL1 (Order Header Register)

• **Condition:** The process will not proceed if the invoice date is invalid.

Special Conditions (Program-Specific)

1. Check for Duplicate Invoice Number (LO101R)

• **Logic:** The program checks if the invoice number has been used before for the supplier. If it has, an error message is displayed.

• **File:** SIHELA (Historical Invoice Register)

• **Field:** C3SFAK

• **Condition:** The process will not proceed if the invoice number is found in the historical register.

2. Check for Valid Project Code (LO101R)

• **Logic:** The program checks if the project code exists in the account register. If not found, it sets an error indicator.

• **File:** RP1PL1 (Project Number Register)

• **Field:** C3PROJ

• **Condition:** The process will not proceed if the project code does not exist.

Subprogram Calls Affecting Logic

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

1. CO402R (Validation of Warehouse)

• **Trigger:** Called during initialization to check if a new warehouse is created.

• **Logic:** Validates if the warehouse differs from the user's default warehouse.

• **Impact:** This call determines if the program should use the overridden warehouse department.

2. RA507R (Department Query)

• **Trigger:** Called when querying department information.

• **Logic:** Retrieves department details based on the department code.

• **Impact:** Ensures that valid department information is displayed and used throughout the process.

3. AA007R (Error Notification)

• **Trigger:** Called whenever an error message needs to be displayed to the user.

• **Logic:** Displays custom error messages based on various validation checks.

• **Impact:** This is a critical step for user interaction, ensuring that users are informed of any issues during the registration process.

