FO100R.MBR

Path: NXKORR/rpgsrc/FO100R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 18014ms

---

Business Logic for Order Registration

This document outlines the business rules that govern the order registration process, based on an analysis of the RPG program FO100R. The primary focus is on the validation and processing of customer orders, including checks for credit limits, order types, and customer details.

The core logic for order registration is contained within the hente_ordre subroutine in FO100R. The program processes customer orders by validating input data, checking customer credit status, and updating order details in the database.

Order Status and Header Rules

Order Registration: FOHEUR, RKUNPF, VOTYL1, VOT1L1, AVALL1

1. Check for Existing Orders

- **Logic:** The program checks if the order number already exists in the order header register. If it does, it prevents the creation of a new order.
- **File:** FOHEUR (Order Header Register)
- **Field:** FOHEUR.NR
- **Condition:** "The process will not allow the creation of a new order if FOHEUR.NR is already in use."

2. Credit Limit Check

- **Logic:** The program checks if the customer has exceeded their credit limit. If the limit is exceeded, the order may still be created if it is a quote (order type 'T').
- **File:** RKUNPF (Customer Register)
- **Field:** RKUNPF.CREDIT_LIMIT
- **Condition:** "The process will not allow the order to be finalized if RKUNPF.CREDIT_LIMIT is exceeded, unless the order type is 'T'."

Configuration and Authorization Rules

1. User Permissions for Customer Creation

- **Logic:** The program checks if the user has permission to create new customers. If not, it displays an error message.
- **Files:**
- FUSRL1 (User Register)
- AVALL1 (Validation Codes)
- **Fields:**
- FUSRL1.USER_ID (from FUSRL1)
- AVALL1.PERMISSION_CODE (from AVALL1)
- **Condition:** "The process will not allow customer creation if the user does not have the required permissions."

2. Order Type Validation

- **Logic:** The program validates that the order type has a system code of 0 (Sales/Ofak). If the order type is not valid, an error is generated.
- **File:** VOTYL1 (Order Type Register)
- **Field:** VOTYL1.SYSTEM_CODE
- **Condition:** "The process will not proceed if VOTYL1.SYSTEM_CODE is not equal to 0."

Financial and Transactional Rules

1. Handling Customer Credit Holds

- **Logic:** If a customer is on credit hold, the program allows the registration of quotes but restricts the ability to create orders until the hold is lifted.
- **File:** RKUNPF (Customer Register)
- **Fields:**
- RKUNPF.CREDIT_HOLD (Customer Credit Hold Status)
- RKUNPF.CREDIT_LIMIT (Customer Credit Limit)
- **Condition:** "The process will not allow order creation if RKUNPF.CREDIT_HOLD is active and the order type is not 'T'."

2. Validation of Delivery Dates

- **Logic:** The program checks if the delivery date is valid and not earlier than the order date. If invalid, an error message is displayed.
- **File:** FOHEUR (Order Header Register)
- **Condition:** "The process will not allow the order to be finalized if the delivery date is earlier than the order date."

Special Conditions (Program-Specific)

1. Requisition Requirement Notification (FO100R)

- **Logic:** If a customer requires a requisition for orders, the program checks if the requisition field is filled. If not, it displays an error message.
- **File:** RKUNPF (Customer Register)
- **Field:** RKUNPF.REQUISITION_REQUIRED
- **Condition:** "The process will not proceed if RKUNPF.REQUISITION_REQUIRED is true and the requisition field is blank."

2. Order Type for Quotes (FO100R)

- **Logic:** The program allows the creation of quotes even if the customer has a credit hold. However, it requires approval for finalizing orders.
- **File:** RKUNPF (Customer Register)
- **Field:** RKUNPF.CREDIT_HOLD
- **Condition:** "The process allows quotes to be created even if RKUNPF.CREDIT_HOLD is active, but requires approval for orders."

Subprogram Calls Affecting Logic

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

1. **FO101R (Order Header Maintenance)**

- **Trigger:** Called when the user wants to maintain order header details.
- **Logic:** This program allows the user to update order header information.
- **Impact:** "This call ensures that the order header is maintained correctly."

2. **FO730R (Order Calculation)**

- **Trigger:** Called when an order is created or updated.
- **Logic:** This program calculates the total order value based on line items.
- **Impact:** "This is a critical step in ensuring accurate order pricing."

3. **AA005R (Error Messaging)**

- **Trigger:** Called when an error condition is met.
- **Logic:** This program displays error messages to the user.
- **Impact:** "This call provides user feedback and ensures that errors are communicated effectively."

This documentation serves as a guide to understanding the business logic implemented in the FO100R program for order registration, detailing the rules, conditions, and subprogram interactions that govern the process.