NN700R.MBR

Path: NXCLOUD/rpgsrc/NN700R.MBR **Generated**: 2026-01-08 12:35:05 **Processing Time**: 10048ms

Business Logic for Customer Information Export

This document outlines the business rules that govern the export of customer information from the online store, based on an analysis of the RPG program NN700R. The primary focus is on the logic that handles the retrieval and formatting of customer data for export.

The core logic for exporting customer information is contained within the main program logic in NN700R. The program retrieves data from various files, formats it according to specified structures, and writes it to an output file.

Customer Data and Record Structure Rules

Customer Information Export: rkunl1, fkprl1, apospf, anbll3, anbhl1, netkund, netkrapp

1. Retrieve Online Store Information

- **Logic:** The program retrieves the online store's firm and group information. If no information is found, the process terminates with an error status.
- **File:** anbll3 (Online Store Information)
- **Field:** anbll3_fgr, anbll3_fir
- **Condition:** The process will not proceed if the online store information is not found.

2. Message Header Creation

- **Logic:** The program creates a message header with a unique reference number and type ID for the export.
- **File:** N/A (In-memory structure)
- **Field:** d_unhrefno, d_unhtypeid
- **Condition:** This step is always executed after retrieving store information.

3. Organization Information Export

- **Logic:** The program exports organization details associated with the online store.
- **File:** anbhl1 (Organization Information)
- **Field:** aolheie, aolbut, aohnav
- **Condition:** This step is executed after the message header is created.

4. Customer Data Processing

- **Logic:** The program processes each customer record, checking for required fields and writing valid records to the output.
- **File:** rkunl1 (Customer Information)
- **Field:** rkebiz, rkemal, rknavn
- **Condition:** The process will skip customers missing mandatory fields like email address or organization number, logging a message for each skipped customer.

5. Customer Project Data Export

- **Logic:** For each customer, the program checks and exports associated project data if the project date is valid.

•**File:** fkprl1 (Customer Project Information)
•**Field:** fltdat, flkund, flkpro
•**Condition:** The project data is only exported if the project date is greater than or equal to the current date.

Error Handling and Logging Rules

1. Error Logging for Missing Organization Number
•**Logic:** If a customer is missing an organization number, a warning message is logged, and the customer is treated as a private customer.
•**File:** N/A (In-memory structure)
•**Field:** SrcDta
•**Condition:** This occurs if rkftnr is zero.
2. Error Logging for Missing Address
•**Logic:** If a customer is missing a postal address, a warning message is logged, and the customer is skipped.
•**File:** N/A (In-memory structure)
•**Field:** SrcDta
•**Condition:** This occurs if the address field is blank.
3. Error Logging for Missing Email Address
•**Logic:** If a customer is missing an email address, a warning message is logged, and the customer is skipped.
•**File:** N/A (In-memory structure)
•**Field:** SrcDta
•**Condition:** This occurs if rkemal is blank for business customers.

Subprogram Calls Affecting Logic

Beyond direct file checks, several external subprograms are called that play a significant role in the workflow.
1. *inzsr (Initialization Subroutine)
•**Trigger:** This subroutine is called at the beginning of the program.
•**Logic:** It initializes key fields and retrieves the firm information from the local data area.
•**Impact:** This setup is crucial for ensuring that the program has the necessary context to retrieve customer and project data.
2. *entry (Entry Point)
•**Trigger:** This is the main entry point for the program.
•**Logic:** It accepts parameters and sets up the environment for processing customer data.
•**Impact:** This establishes the initial state and prepares the program for execution.
3. *return (Termination)
•**Trigger:** This is called at the end of the program.
•**Logic:** It cleans up and sets the last record indicator.
•**Impact:** This ensures that resources are released and the program exits cleanly.

This documentation serves as a comprehensive guide to the business logic implemented in the NN700R program for exporting customer information from the online store. Each rule and condition is designed to ensure data integrity and proper handling of customer records.