LA101R.MBR

Path: NXCLOUD/rpgsrc/LA101R.MBR **Generated**: 2026-01-08 15:04:00 **Processing Time**: 12728ms

Business Logic for LA101R

This document outlines the business rules that govern the maintenance of status codes within the system, based on an analysis of the RPG program LA101R. The primary focus is on how the program handles the retrieval and updating of inventory codes.

The core logic for maintaining inventory codes is contained within the A1BLD subroutine in LA101R. The program processes records from the lstsl1 and lstslu files, which represent the inventory status codes.

Inventory Status Code Management: lstsl1 and lstslu

LA101R: lstsl1, lstslu

1. Retrieve Inventory Code

- **Logic:** The program retrieves details of an inventory code based on a key from the lstsl1 file. If found, it populates various fields with the corresponding values; if not found, it initializes these fields to blank or zero.
- **File:** lstsl1 (Inventory Status Codes)
- **Field:** lstsl1_key
- **Condition:** The process will not select a record if the lstsl1_key is not found.

2. Display Inventory Code

- **Logic:** After retrieving the inventory code, the program displays the information on the screen using the a1bld display format.
- **File:** a1bld (Display Format)
- **Field:** Various fields populated from lstsl1
- **Condition:** The display is triggered after the retrieval of the inventory code.

Update Inventory Code Rules

1. Update Existing Inventory Code

- **Logic:** If the inventory code exists in the lstslu file, the program updates the record with the values retrieved from the lstsl1 file.
- **Files:**
- lstslu (Updated Inventory Status Codes)
- lstsl1 (Original Inventory Status Codes)
- **Fields:**
- laabrk (Status Code)
- laabin (Inventory Number)
- **Condition:** The update occurs only if the lstslu_key is found.

2. Insert New Inventory Code

- **Logic:** If the inventory code does not exist in the lstslu file, the program writes a new record using the values from the lstsl1 file.

•**File:** lstslu (Updated Inventory Status Codes)
•**Field:** laafir (Firm Identifier)
•**Condition:** The insert operation is performed if the lstslu_key is not found.

Special Conditions (Program-Specific)

1. Program Termination Logic (LA101R)

•**Logic:** The program checks for a termination condition based on user input (F3 key). If the termination condition is met, the program sets the last record indicator and returns control to the caller.
•**File:** None
•**Field:** None
•**Condition:** The program will terminate if the user presses the F3 key.

2. Initialization Subroutine (LA101R)

•**Logic:** The initialization subroutine sets up the keys for reading and updating the inventory codes and assigns the firm identifier to the working variable.
•**File:** None
•**Fields:** w_firm (Firm Identifier)
•**Condition:** This subroutine is called at the beginning of the program execution.

Subprogram Calls Affecting Logic

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

1. A1BLD (Display Subroutine)

•**Trigger:** This subroutine is called after retrieving the inventory code details.
•**Logic:** It formats and displays the inventory code information to the user.
•**Impact:** This call acts as a **major logical gateway** for user interaction with the inventory codes.

2. *inzsr (Initialization Subroutine)

•**Trigger:** This subroutine is called at the start of the program.
•**Logic:** It initializes the keys used for reading and updating inventory codes.
•**Impact:** This is a **critical setup step** that ensures the program has the necessary context to operate correctly.

3. end_pgm (Program Termination)

•**Trigger:** This is called when the program is about to terminate.
•**Logic:** It sets the last record indicator and returns control to the calling program.
•**Impact:** This represents the **finalization of the program's execution**, ensuring proper cleanup and state management.