AA000R.MBR

Path: NXCLOUD/rpgsrc/AA000R.MBR **Generated**: 2026-01-08 11:56:13 **Processing Time**: 9166ms

Business Logic for User Registration Lookup

This document outlines the business rules that govern the user registration lookup process, based on an analysis of the RPG program AA000R. The primary focus is on how the program retrieves user information from the user register file and validates the input parameters.

The core logic for user lookup is contained within the main processing logic of the program, which includes reading from the user register file and performing validation checks on the user input.

User Registration and Validation Rules

User Registration Lookup: ausrl1

1. User Record Retrieval

- **Logic:** The program attempts to retrieve a user record based on the provided user identifier (p_user). If the record is found, various user-related fields are populated with data from the retrieved record.
- **File:** ausrl1 (User register file)
- **Field:** ausrl1_user
- **Condition:** The process will not proceed if the user record is not found (*in60 is set to *off).

2. User Existence Validation

- **Logic:** If the user record is not found, the program sets p_okok and dsokok to 0, indicating that the user lookup was unsuccessful.
- **File:** ausrl1 (User register file)
- **Field:** ausrl1_key
- **Condition:** The user record must exist for the process to be considered successful.

Special Conditions (Program-Specific)

1. Web Store User Exception

- **Logic:** For specific users associated with the web store (e.g., 'ASPKASSE', 'NORGROS', 'ASPRMI'), the program does not require a user to be defined. If the user matches these identifiers or starts with 'Q', the lookup is considered unsuccessful.
- **File:** ausrl1 (User register file)
- **Field:** p_user
- **Condition:** If p_user matches 'ASPKASSE', 'NORGROS', 'ASPRMI', or starts with 'Q', the program sets p_okok and dsokok to 0.

2. Initialization Subroutine (AA000R)

- **Logic:** The initialization subroutine sets the initial state of the program, particularly the p_okok variable to 0, indicating that the process has not yet succeeded.
- **File:** ausrl1 (User register file)
- **Field:** p_okok
- **Condition:** This subroutine is called at the start of the program to prepare for user lookup.

Subprogram Calls Affecting Logic

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

1. AA000R (User Registration Lookup Program)

- **Trigger:** This program is executed when a lookup for a user is initiated.
- **Logic:** It checks for a user record in the user register and validates the user input against predefined exceptions.
- **Impact:** This call acts as a **major logical gateway** for user validation and retrieval, determining if the user can be processed further.

2. *inzsr (Initialization Subroutine)

- **Trigger:** This subroutine is automatically called at the beginning of the program.
- **Logic:** It initializes variables and prepares the program for execution.
- **Impact:** This is a **critical setup step** that ensures all necessary variables are in a known state before processing begins.

3. File Operations

- **Trigger:** The program performs file operations to read user records.
- **Logic:** It chains to the user register file to retrieve user data based on the provided user identifier.
- **Impact:** This represents the **core data retrieval function** of the program, essential for the user lookup process.