AA000R.MBR

Path: NXCLOUD/rpgsrc/AA000R.MBR **Generated**: 2026-01-08 12:26:00 **Processing Time**: 9234ms

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 user records are retrieved and validated against specific conditions.

The core logic for user lookup is contained within the main processing logic of the program, which retrieves user records from the ausrl1 file and performs various checks to ensure the validity of the user information.

User Record Retrieval Rules

User Lookup Process: 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.
- **File:** ausrl1 (User Registration File)
- **Field:** ausrl1_user
- **Condition:** The program will not proceed if the user record is not found (*in60 = *off).

2. User Validation for Specific Cases

- **Logic:** The program checks if the user identifier matches certain predefined values or if the first character of the user identifier is 'Q'. If so, it sets the validation status to false.
- **File:** ausrl1 (User Registration File)
- **Field:** p_user
- **Condition:** The validation fails if p_user equals 'ASPKASSE ', 'NORGROS ', 'ASPRMI ', or if the first character of p_user is 'Q'.

Initialization and Parameter Rules

1. Initialization of Parameters

- **Logic:** Upon program entry, the parameters for user validation and job identifier are initialized to default values.
- **Files:** None (local variables)
- **Fields:**
- p_okok (Validation Status)
- p_user (User Identifier)
- p_jobb (Job Identifier)
- **Condition:** This initialization occurs at the start of the program in the initialization subroutine.

2. User Identifier Assignment

- **Logic:** The user identifier is assigned to a local variable for further processing.
- **File:** None (local variable)
- **Field:** ausrl1_user

•**Condition:** This assignment occurs before the record retrieval attempt.

Special Conditions

1. User Record Not Required for Online Users

•**Logic:** For users coming from an online store, there is no requirement for the user to be defined in the system.

•**File:** None (business logic)

•**Field:** p_user

•**Condition:** This condition is checked to allow online users to bypass the normal user validation process.

2. End of Program Logic

•**Logic:** The program concludes by setting the last record indicator to indicate that the program has finished processing.

•**File:** None (control logic)

•**Field:** *inlr

•**Condition:** This occurs at the end of the main processing logic.

Subprogram Calls Affecting Logic

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

1. Initialization Subroutine (AA000R)

•**Trigger:** This subroutine is called at the beginning of the program.

•**Logic:** It initializes the parameters p_okok, p_user, and p_jobb to ensure they have default values before processing.

•**Impact:** This initialization is crucial for ensuring that the program starts with a clean state and avoids processing with undefined values.

2. User Record Chain Logic (AA000R)

•**Trigger:** The user record is retrieved using a chain operation.

•**Logic:** The program checks if the user record exists and populates various fields based on the retrieved data.

•**Impact:** This is a critical step that determines whether the user is valid and what data will be processed in subsequent steps.

3. Validation Logic (AA000R)

•**Trigger:** This logic is executed after attempting to retrieve the user record.

•**Logic:** It checks specific conditions to determine if the user is valid or if they should be flagged as invalid.

•**Impact:** This validation step is essential for maintaining data integrity and ensuring that only authorized users are processed.