AA000R.MBR

Path: NXCLOUD/rpgsrc/AA000R.MBR **Generated**: 2026-01-08 15:03:29 **Processing Time**: 12954ms

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 program logic of AA000R, which processes user data by checking against a user file and applying various validation rules.

User Record Retrieval Rules

User Lookup: ausrl1

1. User Record Retrieval

- **Logic:** The program attempts to retrieve a user record from the ausrl1 file based on the provided user identifier (p_user). If the record is found, various fields are populated with values from the user record.
- **File:** ausrl1 (User registration file)
- **Field:** ausrl1_user
- **Condition:** The process will not select a record if *in60 (the indicator for a successful chain operation) is set to *off, indicating that the user record does not exist.

2. User Validation

- **Logic:** After attempting to retrieve the user record, the program checks if the user is from specific online store identifiers (e.g., 'ASPKASSE', 'NORGROS', 'ASPRMI') or if the first character of p_user is 'Q'. If so, the user is deemed invalid.
- **File:** ausrl1 (User registration file)
- **Field:** p_user
- **Condition:** The process will set p_okok to 0 (indicating failure) if the user matches any of the specified conditions.

Initialization and Parameter Handling Rules

1. Program Initialization

- **Logic:** The program initializes the p_okok parameter to 0 upon entry, indicating that no valid user has been processed yet.
- **Files:** None
- **Fields:** p_okok
- **Condition:** This initialization occurs every time the program is called.

2. Parameter Passing

- **Logic:** The program accepts parameters for user identification (p_user), job identifier (p_jobb), and a status indicator (p_okok). These parameters are used throughout the program for processing.
- **File:** None

•**Fields:** p_user, p_jobb, p_okok

•**Condition:** The parameters must be passed correctly for the program to function as intended.

Special Conditions (Program-Specific)

1. User Record Not Found Handling (AA000R)

•**Logic:** If the user record is not found, the program sets p_okok and dsokok to 0, indicating that the lookup was unsuccessful.

•**File:** ausrl1 (User registration file)

•**Field:** p_okok

•**Condition:** This occurs when *in60 is *off after attempting to chain to the user record.

2. Online Store User Handling (AA000R)

•**Logic:** Users from specific online stores are allowed to bypass the user definition requirement, and the program sets p_okok to 0 if the user matches these identifiers.

•**File:** ausrl1 (User registration file)

•**Fields:** p_user

•**Condition:** This check is performed after the user record retrieval attempt.

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 the p_okok variable to 0 and sets up the parameters for the program.

•**Impact:** This initialization is crucial for ensuring that the program starts with a clean state for processing user records.

2. *entry (Entry Point)

•**Trigger:** This is the entry point of the program where parameters are passed.

•**Logic:** It defines the parameters that the program will use, including p_okok, p_user, and p_jobb.

•**Impact:** Proper parameter handling is essential for the program's functionality, affecting how user data is processed.

3. chain (File Access)

•**Trigger:** The chain operation is invoked to retrieve user records from the ausrl1 file.

•**Logic:** It attempts to find a record based on the ausrl1_user key.

•**Impact:** This operation is a critical step in determining whether a user exists in the system and influences the subsequent validation logic.