

AA000R.MBR

Path: NXCLOUD/rpgsrc/AA000R.MBR **Generated:** 2026-01-08 12:04:32 **Processing Time:** 8841ms

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 AA000R program. The program retrieves user records from the ausrl1 file and checks various conditions to determine if the user is valid for further processing.

User Record Validation Rules

User 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, it populates various data structures with user information.
- **File:** ausrl1 (User Registration 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 Check

- **Logic:** If the user record is not found, the program sets p_okok and dsokok to 0, indicating an unsuccessful lookup.
- **File:** ausrl1 (User Registration File)
- **Field:** ausrl1_user
- **Condition:** Triggered when the user record is not found.

Special Conditions for User Validation

1. Web Store User Exception

- **Logic:** Users from specific web stores (ASPKASSE, NORGROS, ASPRMI) or those whose identifier starts with 'Q' are not required to be defined in the user register.
- **File:** ausrl1 (User Registration File)
- **Field:** p_user
- **Condition:** The process will set p_okok and dsokok to 0 if the user matches any of the specified conditions.

2. Initialization Subroutine

- **Logic:** The program initializes parameters for the user lookup process, setting p_okok to 0 at the start.
- **File:** ausrl1 (User Registration File)
- **Field:** p_okok
- **Condition:** This is executed at the beginning of the program to prepare for user validation.

Subprogram Calls Affecting Logic

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

1. *inzsr (Initialization Subroutine)

- Trigger:** This subroutine is called at the start of the program.
- Logic:** It initializes the parameters for the user lookup process.
- Impact:** This setup is crucial for ensuring that the program starts with a clean state, particularly setting p_okok to 0.

2. chain (File Access Logic)

- Trigger:** The program attempts to chain to the ausrl1 file using the user identifier.
- Logic:** This operation retrieves the user record based on the key.
- Impact:** The success or failure of this operation directly affects the validation logic and subsequent processing steps.

3. eval (Variable Assignment Logic)

- Trigger:** Throughout the program, various evaluations are performed to assign values to local data structures.
- Logic:** This includes populating user data fields if a valid record is found.
- Impact:** The assignment of these values is essential for the program's logic to function correctly, as they are used in later processing steps.

This documentation provides a comprehensive overview of the business logic for the user registration lookup process in the AA000R program, detailing how user records are accessed and validated based on specific conditions.