

AA000R.MBR

Path: NXCLOUD/rpgsrc/AA000R.MBR **Generated:** 2026-01-08 12:16:19 **Processing Time:** 11001ms

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 user input.

The core logic for user registration lookup is contained within the main processing section of the AA000R program. The program checks for the existence of a user record and sets various status indicators based on the results of this lookup.

User Record Lookup Rules

User Registration: ausrl1

1. User Record Existence Check

- **Logic:** The program attempts to retrieve a user record based on the provided user ID. If the record is not found, it sets the status indicators to indicate failure.
- **File:** ausrl1 (User register file)
- **Field:** ausrl1_user
- **Condition:** The process will not proceed if the user record is not found (*in60 = *off).

2. User Status Initialization

- **Logic:** If the user record is found, the program initializes various data structures with user-related information.

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

• **Fields:**

- dsbsid (Job ID)
- dsuser (User ID)
- dsmenty (Menu ID)
- dssfil (File status)
- dssbib (Bibliography status)
- dssfir (Firm status)
- dssavd (Department status)
- dsfnav (Navigation status)
- dsbfil (Backup file status)
- dsbfir (Backup firm status)
- dsbavd (Backup department status)
- dslevl (Level status)
- dslagr (Lager status)
- dsmilj (Environment status)

• **Condition:** This initialization occurs only if the user record is successfully retrieved.

User Validation Rules

1. User Input Validation

- **Logic:** The program checks if the user ID matches specific values or if the first character of the user ID is 'Q'. If any of these conditions are met, the process will not proceed.
- **File:** ausrl1 (User register file)
- **Field:** p_user
- **Condition:** The process will not proceed if p_user equals 'ASPKASSE ', 'NORGROS ', 'ASPRMI ', or if the first character of p_user is 'Q'.

-

Special Conditions (Program-Specific)

1. Program Termination Logic

- **Logic:** The program sets the last record indicator (*inlr) to on, signaling the end of the program execution.
- **File:** N/A
- **Condition:** This action occurs at the end of the program after all processing is complete.

2. Program Initialization Subroutine

- **Logic:** The program initializes parameters upon entry, setting p_okok to 0 to indicate that the process has not yet succeeded.
- **File:** N/A
- **Condition:** This initialization occurs at the start 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. AA000R (User Registration Lookup Program)

- **Trigger:** This program is called when a user needs to be validated against the user register.
- **Logic:** It retrieves user information based on the user ID and initializes various status fields.
- **Impact:** This call acts as a **major logical gateway** for user validation and ensures that only valid users are processed.

2. *inzsr (Initialization Subroutine)

- **Trigger:** This subroutine is called at the beginning of the program to set up initial parameters.
- **Logic:** It initializes the p_okok parameter to indicate the success status of the lookup process.
- **Impact:** This is a **critical setup step** that prepares the program for subsequent logic execution.

This documentation provides a comprehensive overview of the business logic implemented in the AA000R program, focusing on user registration lookup and validation processes.