AA000R.MBR

Path: NXCLOUD/rpgsrc/AA000R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 11899ms

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 accessed and validated within the system.

The core logic for user registration lookup is contained within the main program logic of AA000R. The program retrieves user records from the ausrl1 file and performs validation checks based on the input parameters.

User Record Access and Validation Rules

User Registration Lookup: ausrl1

1. Retrieve User Record

- **Logic:** The program attempts to retrieve a user record from the ausrl1 file using the provided user ID.
- **File:** ausrl1 (User registration file)
- **Field:** ausrl1_user
- **Condition:** The record is retrieved if the user ID exists in the ausrl1 file; if not, the process sets p_okok and dsokok to 0.

2. Check for Valid User

- **Logic:** If the user ID is either 'ASPKASSE', 'NORGROS', 'ASPRMI', or starts with 'Q', the user is considered invalid.
- **File:** ausrl1 (User registration file)
- **Field:** ausrl1_user
- **Condition:** The process will set p_okok and dsokok to 0 if the user ID matches any of the specified invalid conditions.

Initialization and Parameter Handling Rules

1. Initialize Program Parameters

- **Logic:** Upon entry into the program, the parameters p_okok, p_user, and p_jobb are initialized to prepare for processing.
- **Files:**
- ausrl1 (User registration file)
- **Fields:**
- p_okok (Status flag)
- p_user (User ID)
- **Condition:** This initialization occurs at the start of the program execution.

2. Set User Data Fields

- **Logic:** If the user record is successfully retrieved, various data fields are populated with user information from the ausrl1 file.

•**File:** ausrl1 (User registration file)
•**Fields:**
•dswsid (Job ID)
•dsuser (User ID)
•dsmeny (Menu ID)
•dssfil (File ID)
•dssbib (Bib ID)
•dssfir (Firm ID)
•dssavd (Department ID)
•dsfnav (Navigation ID)
•dsbfil (Backup File ID)
•dsbfir (Backup Firm ID)
•dsbavd (Backup Department ID)
•dslevl (Level)
•dslagr (Lagr)
•dsmilj (Miljø)
•**Condition:** This assignment occurs only if the user record is found successfully.

Special Conditions (Program-Specific)

1. User from Webshop
•**Logic:** Users coming from the webshop do not need to be defined in the user registration file.
•**File:** ausrl1 (User registration file)
•**Field:** p_user
•**Condition:** The program allows for users whose ID starts with 'Q' to bypass the standard validation checks.
2. Program Termination
•**Logic:** The program sets the last record indicator (*inlr) to on to terminate the program.
•**File:** N/A
•**Condition:** This occurs at the end 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. *inzsr (Initialization Subroutine)
•**Trigger:** This subroutine is called at the beginning of the program.
•**Logic:** It initializes the program parameters and prepares the environment for processing.
•**Impact:** This call ensures that the program starts with a clean state, setting up necessary flags and parameters.
2. *entry (Entry Point)
•**Trigger:** This is where the program begins processing with input parameters.
•**Logic:** It accepts parameters for user ID and job ID, which are crucial for the user lookup process.

•**Impact:** This establishes the context for the user lookup and validation logic.

3. chain (File Access Logic)

•**Trigger:** The chain operation is called to access the user record.

•**Logic:** It attempts to find the user record in the ausrl1 file based on the user ID.

•**Impact:** This is a critical step as it determines whether the user record exists and affects the subsequent validation logic.