FA120R.MBR

Path: NXCLOUD/rpgsrc/FA120R.MBR **Generated**: 2026-01-08 12:06:12 **Processing Time**: 10424ms

Business Logic for Job Register Status Maintenance

This document outlines the business rules that govern the Job Register Status Maintenance process, based on an analysis of the RPG program FA120R. The primary focus is on how job register records are retrieved, displayed, and updated within the system.

The core logic for maintaining job register status is contained within the *inzsr subroutine in FA120R. The program processes records from the job register and user register files, allowing for updates and retrieval of job-related information.

Job Register and User Management Rules

Job Register Maintenance: fjobl1, fjoblu, fausrl1

1. Retrieve Job Information

- **Logic:** The program retrieves job details from the job register file (fjobl1). If the record is found, it populates various job fields; if not found, it initializes them to zero.
- **File:** fjobl1 (Job Register File)
- **Field:** fajb01 to fajb10
- **Condition:** The process will not select a record if the key lookup for fjobl1 fails (*in90 is on).

2. Display Job Information

- **Logic:** After retrieving job information, the program displays it on the screen using the a1bld format.
- **File:** ffa120d (Display File)
- **Field:** Various job fields (e.g., a1jb01, a1jb02, etc.)
- **Condition:** The display occurs after successful retrieval of job data.

User Query and Interaction Rules

1. User Lookup for Job

- **Logic:** When querying who started a job, the program checks the specified field and retrieves the corresponding user information from the user register (ausrl1).
- **Files:**
- fausrl1 (User Register File)
- **Fields:**
- ausrl1_user (User ID)
- **Condition:** The user lookup occurs based on the field specified in w_afld (e.g., A1JB01, A1JB02 , etc.).

2. Job Update Logic

- **Logic:** The program updates the job register with the current user information and timestamps. If the job record already exists, it updates; otherwise, it creates a new record.
- **File:** fjoblu (Job Update File)

- **Field:** fajb01 to fajb10
- **Condition:** The update occurs if the job record is found (*in90 is off).

Special Conditions (Program-Specific)

1. Initialization of Program (FA120R)

- **Logic:** The program initializes keys for reading and updating job and user registers, setting the firm number in the key.
- **File:** fjobl1, fjoblu, ausrl1
- **Field:** w_firm, ausrl1_user
- **Condition:** This initialization is required before any data processing can occur.

2. End of Program Logic

- **Logic:** The program sets the last record indicator to on and returns control to the calling program.
- **File:** None
- **Condition:** This occurs at the end of the processing logic.

Subprogram Calls Affecting Logic

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

1. x_user (User Lookup Subroutine)

- **Trigger:** Called when a user needs to be looked up based on job fields.
- **Logic:** This subroutine retrieves the user's name from the user register based on the user ID.
- **Impact:** This call ensures that the job records are associated with the correct user information.

2. *inzsr (Initialization Subroutine)

- **Trigger:** Called at the beginning of the program to set up keys.
- **Logic:** Initializes keys for job and user registers and sets the firm number.
- **Impact:** This is a crucial setup step that allows the program to function correctly.

3. exfmt (Display Subroutine)

- **Trigger:** Called to display job information on the screen.
- **Logic:** This subroutine formats and presents the job data to the user.
- **Impact:** This provides a user interface for interacting with job data, allowing for visibility and updates.

This documentation provides a comprehensive overview of the business logic implemented in the FA120R program, detailing how job registers are maintained and user interactions are handled.