FA120R.MBR

Path: NXCLOUD/rpgsrc/FA120R.MBR **Generated**: 2026-01-08 15:43:00 **Processing Time**: 17352ms

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 status records are retrieved, displayed, and updated within the system.

The core logic for managing job register statuses is contained within the *inzsr subroutine in FA120R. The program retrieves job status information from the job register file, presents it to the user, and allows for updates based on user interactions.

Job Status and Header Rules

Job_Register_Status: fjobl1, fjoblu, fausrl1

1. Retrieve Job Status

•**Logic:** The program retrieves job status information from the fjobl1 file based on a key. If the record is found, it populates several job status fields; if not, it initializes these fields to zero.

•**File:** fjobl1 (Job Register File)

•**Field:** Various fields (fajb01 to fajb10)

•**Condition:** The process will not select a record if the key lookup fails (*in90 is set to *on).

2. Display Job Status

•**Logic:** After retrieving the job status, the program displays the information on the screen using the a1bld format.

•**File:** ffa120d (Display File)

•**Field:** N/A

•**Condition:** The display occurs after the job status is successfully retrieved.

Configuration and Authorization Rules

1. User Lookup

•**Logic:** The program checks the user register (ausrl1) to retrieve the user's name based on the user ID associated with the job status.

•**Files:**

•fausrl1 (User Register File)

•**Fields:**

•ausrl1_user (User ID from Job Status)

•**Condition:** The user lookup is performed if the job status field is accessed.

2. Job Status Update

•**Logic:** When updating the job status, the program sets various fields based on the current job status values and updates the record in the fjoblu file.

•**File:** fjoblu (Job Register Update File)

•**Field:** Various fields (fajb01 to fajb10)

•**Condition:** The update occurs if the record is found (*in90 is set to *off).

Financial and Transactional Rules

1. Job Status Time Stamps

•**Logic:** The program records timestamps for job status updates, capturing the time when the job was updated or created.

•**File:** fjoblu (Job Register Update File)

•**Fields:**

•fajeda (End Date)

•fajeti (End Time)

•**Condition:** Timestamps are recorded only if the record is being updated.

2. Initial Job Status Creation

•**Logic:** If a new job status is being created (record not found), the program initializes the firm number and sets the timestamps.

•**File:** fjoblu (Job Register Update File)

•**Condition:** This occurs when the job status record is not found during the chain operation.

Special Conditions (Program-Specific)

1. Job Status Initialization (FA120R)

•**Logic:** The program initializes keys for the job register and user register at the start of the program.

•**File:** fjobl1, fjoblu, fausrl1 (Job Register and User Register Files)

•**Field:** N/A

•**Condition:** This initialization occurs at the beginning of the program to set up the necessary keys for processing.

2. User Interaction Handling (FA120R)

•**Logic:** The program checks for user inputs (e.g., function keys) to either exit the program or perform specific queries on job statuses.

•**File:** ffa120d (Display File)

•**Fields:** N/A

•**Condition:** User actions trigger different branches of logic, such as exiting or querying job status details.

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 ID is accessed for job status.

•**Logic:** Looks up the user name in the user register based on the user ID.

•**Impact:** This call ensures that the job status displays the correct user information associated with the job.

2. *inzsr (Initialization Subroutine)

- **Trigger:** Called at the start of the program to set up keys.
- **Logic:** Initializes keys for job and user registers.
- **Impact:** This is crucial for ensuring that the program can correctly access and update job status records.

3. exfmt (Display Format Subroutine)

- **Trigger:** Called to display the job status information.
- **Logic:** Renders the job status on the screen for user interaction.
- **Impact:** This represents the primary user interface for interacting with job statuses, allowing users to view and update records.