FA120R.MBR

Path: NXCLOUD/rpgsrc/FA120R.MBR **Generated**: 2026-01-08 11:56:25 **Processing Time**: 11453ms

Business Logic for Job Register Status Maintenance

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

The core logic for managing job statuses is contained within the *inzsr subroutine in FA120R. The program processes records from the job register files, retrieves user information, and updates job statuses based on user interactions.

Job Register and Status Rules

Job Register Maintenance: fjobl1, fjoblu, fausrl1, ffa120d

1. Retrieve Job Status

•**Logic:** The program retrieves job status information from the job register file (fjobl1). If the record is found, it populates various status fields; otherwise, it sets them to zero.

•**File:** fjobl1 (Job Register)

•**Field:** fajb01 to fajb10

•**Condition:** The process will not select a record if the chain operation (*in90) indicates that no record was found.

2. Display Job Status

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

•**File:** ffa120d (Display File)

•**Field:** Various job status fields

•**Condition:** The display is executed after the job status is retrieved and populated.

User Interaction and Query Rules

1. Query Job Starter

•**Logic:** If the user requests to query who started the job, the program checks which field to search based on user input and retrieves the corresponding user information.

•**Files:**

•fausrl1 (User Register)

•**Fields:**

•ausrl1_user (User ID)

•**Condition:** The query is triggered when the user presses the appropriate function key (F1).

2. Update Job Register

•**Logic:** The program updates the job register with the current status values. If the record exists, it updates the existing record; otherwise, it writes a new record.

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

•**Field:** fajb01 to fajb10

•**Condition:** The update occurs if the record is found (*in90 is off) or a new record is created if not found.

Special Conditions (Program-Specific)

1. User Lookup (FA120R)

•**Logic:** The program looks up user information from the user register based on the user ID associated with the job status.

•**File:** fausrl1 (User Register)

•**Field:** ausrl1_user

•**Condition:** The lookup is performed when the x_user subroutine is called.

2. Initialization (FA120R)

•**Logic:** The program initializes keys for accessing job register and user register files, setting the firm number in the key.

•**Files:**

•fjobl1 (Job Register)

•fjoblu (Job Register Update)

•fausrl1 (User Register)

•**Fields:**

•w_firm (Firm Number)

•**Condition:** Initialization occurs at the start of the program.

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 querying user information for a job.

•**Logic:** This subroutine retrieves user details from the user register based on the user ID.

•**Impact:** This call is essential for displaying who started the job and enhances the user experience by providing relevant information.

2. *inzsr (Initialization Subroutine)

•**Trigger:** Automatically invoked when the program starts.

•**Logic:** Initializes keys for file access and sets up necessary variables for processing.

•**Impact:** This subroutine establishes the context for the program's operations, ensuring that all necessary keys are set before any file operations are performed.

3. exfmt (Display Format Subroutine)

•**Trigger:** Called after retrieving job status information.

•**Logic:** Displays the job status information on the screen.

•**Impact:** This is a critical step in the user interface, allowing users to view job statuses interactively.

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