

FA120R.MBR

Path: NXCLOUD/rpgsrc/FA120R.MBR **Generated:** 2026-01-08 12:22:53 **Processing Time:** 12267ms

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 statuses are retrieved, displayed, and updated within the system.

The core logic for job status maintenance is contained within the *inzsr subroutine in FA120R. The program processes job records by retrieving existing job information, allowing user interaction, and updating job statuses based on user input.

Job Status and Record Rules

Job_Register_Status: fjobl1, fjoblu, fausrl1, ffa120d

1. Retrieve Job Information

• **Logic:** The program retrieves job details from the fjobl1 file based on a specified key. If the record is found, it populates various job status fields; otherwise, it initializes them to zero.

• **File:** fjobl1 (Job Register File)

• **Field:** fajb01 to fajb10

• **Condition:** The process will not select a record if the chain operation indicates that the record is not found (*in90 = *off).

2. Display Job Status

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

• **File:** ffa120d (Display File)

• **Field:** Various job status fields

• **Condition:** The display is executed unconditionally after job data retrieval.

User Interaction and Query Rules

1. User Query on Job Initiation

• **Logic:** If the user requests to query who started the job, the program checks the field specified by the user and retrieves the corresponding user information from the ausrl1 file.

• **Files:**

• ausrl1 (User Register File)

• **Fields:**

• ausrl1_user (User ID)

• **Condition:** The query is triggered if the user presses the F1 key (*inka = *on).

2. Field-Specific User Lookup

• **Logic:** Depending on the field selected by the user (e.g., A1JB01 to A1JB10), the program retrieves the corresponding user and timestamp information from the job record.

• **File:** ausrl1 (User Register File)

• **Fields:** ausrl1_user, fada01 to fada10 (Date), fati01 to fati10 (Time)

- **Condition:** The lookup is executed based on the specific field selected by the user.

Job Update Rules

1. Update Job Information

- **Logic:** The program updates the job record in fjoblu with the retrieved job status information. If the record does not exist, it creates a new entry.

- **File:** fjoblu (Job Update File)

- **Fields:** fajb01 to fajb10

- **Condition:** The update occurs if the chain operation indicates that the record was not found ($*in90 = *off$).

2. Set User and Timestamp on Update

- **Logic:** When updating the job record, the program sets the user ID and timestamps for creation and modification.

- **File:** fjoblu (Job Update File)

- **Condition:** This logic executes unconditionally as part of the update process.

Special Conditions (Program-Specific)

1. User Lookup Subroutine (FA120R)

- **Logic:** A subroutine (x_user) is called to perform a lookup in the user register to retrieve the user's name based on the user ID.

- **File:** ausrl1 (User Register File)

- **Field:** ausrl1_user

- **Condition:** The lookup occurs if the user ID is valid ($*in91 = *off$).

2. Initialization Subroutine (FA120R)

- **Logic:** The initialization subroutine sets up the keys for the job register and user register files, ensuring that the firm number is included in the key.

- **Files:**

- fjobl1 (Job Register File)

- fjoblu (Job Update File)

- ausrl1 (User Register File)

- **Condition:** This setup occurs at the start of the program to prepare for data processing.

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 the user queries who started the job.

- **Logic:** This subroutine retrieves the user's name from the user register based on the user ID.

- **Impact:** This call ensures that the job record displays accurate user information.

2. *inzsr (Initialization Subroutine)

- **Trigger:** Called at the beginning of the program to set up necessary keys.

- **Logic:** Initializes keys for accessing job and user registers.

- Impact:** This is a foundational step that prepares the program for subsequent operations.

3. exfmt (Display Subroutine)

- Trigger:** Called to display the job status after retrieval.

- Logic:** This subroutine formats and presents the job information to the user.

- Impact:** Represents the user interface interaction, allowing users to view job statuses effectively.