FA120R.MBR

Path: NXCLOUD/rpgsrc/FA120R.MBR **Generated**: 2026-01-08 12:21:24 **Processing Time**: 13215ms

Business Logic for Job Registration Status Maintenance

This document outlines the business rules that govern the job registration status maintenance process, based on an analysis of the RPG program FA120R. The primary focus is on how job registration records are retrieved, displayed, and updated.

The core logic for job registration status maintenance is contained within the *inzsr subroutine in FA120R. The program processes records from the job registration file, retrieves user information, and updates job status based on user interactions.

Job Registration and Status Rules

Job Registration Maintenance: fjobl1, fjoblu, fausrl1

1. Retrieve Job Information

- **Logic:** The program retrieves job details from the fjobl1 file based on a key and populates various fields with job data.
- **File:** fjobl1 (Job Registration File)
- **Field:** fajb01 to fajb10
- **Condition:** The process will not select a record if the chain operation returns a status indicating that the record does not exist (i.e., *in90 is *off).

2. Initialize Job Fields

- **Logic:** If no job record is found, all job fields are initialized to zero.
- **File:** fjobl1 (Job Registration File)
- **Field:** fajb01 to fajb10
- **Condition:** This occurs when the chain operation does not find a matching record.

User Lookup and Interaction Rules

1. User Information Retrieval

- **Logic:** The program retrieves user information from the ausrl1 file based on the user key and populates the user name field.
- **Files:**
- ausrl1 (User Registration File)
- **Fields:**
- ausrl1_user (User ID)
- abnavn (User Name)
- **Condition:** The user information is retrieved only if the chain operation on ausrl1 does not indicate an error (i.e., *in91 is *off).

2. Job Start Query

- **Logic:** If the user presses the F1 key, the program queries who started the job and retrieves relevant user details.

•**File:** fjobl1 (Job Registration File)
•**Field:** ausrl1_user
•**Condition:** This action is triggered by the F1 key being pressed (i.e., *inka is *on).

Job Update Rules

1. Update Job Registration

•**Logic:** The program updates the job registration record with the current job status and user information if the record exists; otherwise, it creates a new record.
•**File:** fjoblu (Job Update File)
•**Fields:**
•fajb01 to fajb10 (Job Status Fields)
•**Condition:** The update occurs if the chain operation on fjoblu indicates that the record exists (i.e., *in90 is *off).

2. Set User and Time Stamps

•**Logic:** The program sets the user and timestamp fields when updating or creating a job record.
•**File:** fjoblu (Job Update File)
•**Condition:** The timestamps are set only when a new record is created or an existing record is updated.

Special Conditions (Program-Specific)

1. EDI Locking Mechanism (FA120R)

•**Logic:** The program sets a lock during EDI reading to prevent concurrent modifications.
•**File:** fjobl1 (Job Registration File)
•**Field:** fajb01
•**Condition:** This locking occurs specifically during EDI processing as indicated in the program comments.

2. Compello Overflow Handling (FA120R)

•**Logic:** The program includes logic to handle overflow scenarios when interfacing with Navision.
•**File:** fjobl1 (Job Registration File)
•**Fields:** fajb01 to fajb10
•**Condition:** This handling is triggered under specific conditions noted in the program comments.

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 user information needs to be retrieved based on the user ID.
•**Logic:** This subroutine chains to the ausrl1 file to obtain the user's name.
•**Impact:** This call ensures that the program has the correct user information for display and updates.

2. *inzsr (Initialization Subroutine)

•**Trigger:** Called at the start of the program to set up key fields for job registration.

•**Logic:** This subroutine initializes keys for both job registration and user registration files.
•**Impact:** This is crucial for ensuring that the program can correctly access and manipulate job and user records.

3. exfmt (Display Format Subroutine)

•**Trigger:** Called to display the job registration status screen.
•**Logic:** This subroutine formats and displays the job registration information to the user.
•**Impact:** This represents the primary user interface interaction point for job status maintenance.