FA120R.MBR

Path: NXCLOUD/rpgsrc/FA120R.MBR **Generated**: 2026-01-08 12:04:43 **Processing Time**: 11667ms

Business Logic for Job Register Maintenance

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

The core logic for job register maintenance is contained within the *inzsr subroutine in FA120R. This subroutine initializes the program and sets up the necessary keys for file access, allowing for the retrieval and updating of job records.

Job Status and Record Rules

Job Register Maintenance: fjobl1, fjoblu, fausrl1, ffa120d

1. Retrieve Job Record

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

•**File:** fjobl1 (Job Register)

•**Field:** fajb01 to fajb10

•**Condition:** The process will not select a record if the key lookup does not return a valid record (*in90 = *off).

2. Display Job Information

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

•**File:** ffa120d (Display File)

•**Field:** Various job fields (a1jb01 to a1jb10)

•**Condition:** The display is executed regardless of whether the job record was found or not.

User and Authorization Rules

1. User Lookup

•**Logic:** The program looks up the user associated with the job by chaining to the ausrl1 file. If the user is found, their name is retrieved.

•**Files:**

•fausrl1 (User Register)

•**Fields:**

•ausrl1_user (User ID)

•**Condition:** The user lookup is performed only if a specific action is triggered (e.g., pressing F1).

2. Job Update Authorization

•**Logic:** The program updates the job record in fjoblu based on the information gathered. If the record was not previously found, it writes a new record.

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

•**Field:** fajb01 to fajb10

•**Condition:** The update occurs only if the job record is not found (*in90 = *off).

Transactional Rules

1. Job Record Update

•**Logic:** The program updates the job fields with the new values and timestamps when saving changes to the job record.
•**File:** fjoblu (Job Update Register)
•**Fields:**
•fajeda (End Date)
•fajeti (End Time)
•**Condition:** The update occurs if the job record is found; otherwise, a new record is created.

2. Job Record Creation

•**Logic:** If the job record does not exist, the program creates a new record with the current user and firm information.
•**File:** fjoblu (Job Update Register)
•**Condition:** A new record is created if the initial chain to fjoblu does not find an existing record (*in90 = *off).

Special Conditions (Program-Specific)

1. EDI Locking Mechanism (FA120R)

•**Logic:** The program sets a lock during EDI reading to prevent concurrent modifications.
•**File:** fjoblu (Job Update Register)
•**Field:** fajb01 to fajb10
•**Condition:** This locking mechanism is triggered during specific EDI processing steps.

2. Compello Overflow Locking (FA120R)

•**Logic:** The program sets a lock when handling overflow from the Compello system to ensure data integrity.
•**File:** fjoblu (Job Update Register)
•**Fields:** fajb01 to fajb10
•**Condition:** This check is specific to the integration with the Compello system.

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 actions.
•**Logic:** This subroutine retrieves the user's name from the user register.
•**Impact:** This call ensures that job records are associated with the correct user, enhancing accountability.

2. *inzsr (Initialization Subroutine)

•**Trigger:** Called at the start of the program to set up keys for file access.
•**Logic:** Initializes keys for job and user records, setting the firm number.

•**Impact:** This setup is crucial for ensuring that subsequent operations can correctly access the necessary data.

3. exfmt (Display Format Subroutine)

•**Trigger:** Called to display job information on the screen.

•**Logic:** This subroutine formats and presents job details to the user.

•**Impact:** It represents the primary interface for user interaction with job records, making it a critical component of the process.