AN060R.MBR

Path: NXCLOUD/rpgsrc/AN060R.MBR **Generated**: 2026-01-08 12:22:44 **Processing Time**: 13165ms

Business Logic for AN060R

This document outlines the business rules that govern the processing of historical records during a night job, based on an analysis of the RPG program AN060R. The primary focus is on the rules for managing and displaying historical job records.

The core logic for managing historical records is contained within the *inzsr subroutine in AN060R . The program primarily interacts with the ahstpfr file to read, update, and delete historical records, while utilizing subfile structures for user interaction.

Order Status and Header Rules

AN060R: ahstpfr

1. Record Maintenance

•**Logic:** The program allows for the maintenance of historical records, enabling updates to the status of night jobs.

•**File:** ahstpfr (Historical job records)

•**Field:** ahstl1_dato

•**Condition:** The process will not select a record if sfrn01 is equal to zero.

2. Display Subfile Records

•**Logic:** The program displays records in a subfile format, allowing users to view multiple historical records at once.

•**File:** b1sfl (Subfile for displaying historical records)

•**Field:** b1dato

•**Condition:** The display will only occur if srrn01 is not equal to zero.

Configuration and Authorization Rules

1. Input Validation for Historical Records

•**Logic:** The program checks if a valid selection has been made in the subfile before proceeding with any operations.

•**Files:**

•b1sfl (Subfile for displaying historical records)

•**Fields:**

•b1valg (User selection)

•**Condition:** The operation will only proceed if b1valg is not blank.

2. Record Deletion Authorization

•**Logic:** The program prompts the user for confirmation before deleting a historical record.

•**File:** ahstpfr (Historical job records)

•**Field:** ahstl1_dato

•**Condition:** The deletion process is initiated if the user selects the option to delete (indicated by b1valg being equal to '4').

Financial and Transactional Rules

1. Update Historical Job Status

- **Logic:** The program updates the status of a historical job to indicate it has failed, based on user input.
- **File:** ahstpfr (Historical job records)
- **Fields:**
- ahstl1_dato (Date of the historical record)
- ahstl1_ttid (Job ID)
- **Condition:** The update occurs if the user selects the option to mark a job as failed (indicated by b1valg being equal to '7').

2. Record Existence Check

- **Logic:** The program checks if a record exists before attempting to update it.
- **File:** ahstpfr (Historical job records)
- **Condition:** The program will only update the record if it is found in the file (checked with %found(ahstlu)).

Special Conditions (Program-Specific)

1. Subfile Initialization (AN060R)

- **Logic:** The program initializes the subfile and prepares it for user interaction.
- **File:** b1sfl (Subfile for displaying historical records)
- **Field:** srrn01
- **Condition:** The subfile is cleared and prepared when the program starts.

2. Data Conversion for Display (AN060R)

- **Logic:** The program converts date formats from file to screen format for user display.
- **File:** ahstpfr (Historical job records)
- **Fields:** d_fdat (File date), d_sdat (Screen date)
- **Condition:** This conversion occurs during the display process to ensure the correct format is shown to the user.

Subprogram Calls Affecting Logic

Beyond direct file checks, several external subprograms are called that play a significant role in the workflow.

1. xdsp01 (Display Subfile)

- **Trigger:** This subroutine is called to display the contents of the subfile.
- **Logic:** It checks if there are records to display and formats the subfile for user interaction.
- **Impact:** This call acts as a **major logical gateway** for user interaction with historical records.

2. xdatsf (Data Conversion Subroutine)

- **Trigger:** This subroutine is called to convert dates from file format to screen format.
- **Logic:** It handles the conversion of date fields for proper display.
- **Impact:** This ensures that users see dates in a familiar format, enhancing usability.

3. xd1win (Delete Confirmation Window)

- **Trigger:** This subroutine is called when a user selects to delete a record.
- **Logic:** It manages the display of a confirmation window for deletion.
- **Impact:** This represents the handoff to the next major business function of confirming deletions, ensuring that users do not accidentally delete records.