AN060R.MBR

Path: NXCLOUD/rpgsrc/AN060R.MBR **Generated**: 2026-01-08 15:03:33 **Processing Time**: 13943ms

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 logic related to querying and maintaining historical records.

The core logic for handling historical records is contained within the main processing loop of the AN060R program. This program is designed to manage records related to night jobs, allowing for querying, updating, and deleting historical data.

Order Status and Header Rules

AN060R: ahstl1, ahstlu, fan060d

1. Record Querying

•**Logic:** The program checks if a specific record in the historical file exists based on user input. If the record is found, it retrieves the corresponding details for display.

•**File:** ahstl1 (Historical records file)

•**Field:** ahdato, ahttid

•**Condition:** The process will not select a record if the b2dato (input date) is zero.

2. Record Deletion

•**Logic:** The program allows users to delete a selected historical record. If the user confirms deletion, the record is removed from the database.

•**File:** ahstl1 (Historical records file)

•**Field:** ahstlur

•**Condition:** The deletion is triggered when the user selects option '4' from the menu.

Configuration and Authorization Rules

1. Record Update Authorization

•**Logic:** The program checks if a night job has failed and allows for updating the status to 'OK'. This ensures that the historical record reflects the correct job status.

•**Files:**

•ahstl1 (Historical records file)

•ahstlu (Update records file)

•**Fields:**

•ahstl1_dato (Date of the historical record)

•ahstl1_ttid (Time ID of the historical record)

•**Condition:** This update occurs when the user selects option '7' and the corresponding record is found in ahstlu.

2. Subfile Management

•**Logic:** The program manages a subfile that displays historical records. It allows users to navigate through records, refresh the view, and handle user selections.

•**File:** b1sfl (Subfile for displaying records)
•**Field:** srrn01 (Relative record number in subfile)
•**Condition:** The subfile is populated and refreshed based on user actions and selections.

Financial and Transactional Rules

1. Record Creation

•**Logic:** When a new historical record is created, the program initializes the necessary fields and prepares the record for insertion into the database.
•**File:** ahstl1 (Historical records file)
•**Fields:**
•ahstl1_dato (Date of the historical record)
•ahstl1_ttid (Time ID of the historical record)
•**Condition:** This occurs when the user selects the option to create a new record.

2. Error Handling

•**Logic:** The program includes error handling to notify users of invalid inputs or actions, ensuring that the user experience is smooth and informative.
•**File:** b1sfl (Subfile for displaying records)
•**Condition:** Error messages are displayed when the user attempts to perform actions with invalid data.

Special Conditions (Program-Specific)

1. Job Failure Acknowledgment (AN060R)

•**Logic:** The program allows users to acknowledge a job failure by updating the historical record to reflect the status change.
•**File:** ahstlu (Update records file)
•**Field:** ahtxt1
•**Condition:** This is executed when the user selects option '7' and the record is found.

2. Subfile Refresh Logic (AN060R)

•**Logic:** The program refreshes the subfile display based on user navigation and selections, ensuring that the latest data is presented.
•**File:** b1sfl (Subfile for displaying records)
•**Fields:** b1dato, b1ttid (Date and Time ID of the record)
•**Condition:** This is triggered by user actions such as selecting options to refresh or navigate through records.

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 after it has been populated.
•**Logic:** It formats and presents the subfile data to the user.

•**Impact:** This call acts as a **major logical gateway** for user interaction with the historical records.

2. xdatsf (Data Conversion Subroutine)

•**Trigger:** This subroutine is invoked to convert date formats between the screen and file.

•**Logic:** It handles the conversion of date fields to ensure consistency.

•**Impact:** This is crucial for maintaining data integrity across the application.

3. xclr01 (Clear Subfile)

•**Trigger:** This subroutine is called to clear the subfile before populating it with new data.

•**Logic:** It resets the subfile indicators and prepares it for fresh data input.

•**Impact:** This ensures that the subfile does not display stale or irrelevant data, enhancing user experience.