AN060R.MBR

—

Business Logic for AN060R

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

The core logic for managing historical records is contained within the *inzsr subroutine in AN060R . The program interacts with various files and subfiles to display, update, and delete historical job records based on user input.

—

Order Status and Header Rules

AN060R: ahstpfr, fan060d

1. Display Historical Records

•**Logic:** The program displays historical job records in a subfile format for user interaction.

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

•**Field:** ahdato

•**Condition:** The subfile is populated with records from ahstpfr using the xcrt01 subroutine.

2. Delete Historical Record

•**Logic:** Allows the user to delete a selected historical job record.

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

•**Field:** ahstlur

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

—

Configuration and Authorization Rules

1. Validate User Input

•**Logic:** Checks if the user has made a valid selection in the subfile before proceeding with actions.

•**Files:**

•b1sfl (Subfile for displaying records)

•ahstpfr (Historical job records file)

•**Fields:**

•b1valg (User selection)

•b1dato (Selected date)

•**Condition:** The program checks if b1valg is not blank and if b1dato is zero to validate user input.

2. Update Historical Record Status

•**Logic:** Updates the status of a historical job record to indicate it has failed.

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

•**Field:** ahstlur

•**Condition:** This action is performed when the user selects option '7' to indicate a job failure.

—

Financial and Transactional Rules

1. Record Job Completion

•**Logic:** Records the completion status of a nightly job by updating the historical records.

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

•**Fields:**

•ahstl1_dato (Date of the job)

•ahstl1_ttid (Job type ID)

•**Condition:** This is executed when the user confirms the completion of a job.

2. Notify on Invalid Account

•**Logic:** Alerts the user if an invalid account is selected for a job.

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

•**Condition:** The program triggers an error message if b2dato is zero when a selection is made.

–

Special Conditions (Program-Specific)

1. Handle Deletion Confirmation (AN060R)

•**Logic:** Manages the confirmation dialog for deleting a historical job record.

•**File:** ahstlur (Temporary historical record)

•**Field:** ahstlur

•**Condition:** The confirmation is prompted when the user selects option '4' to delete a record.

 Note: This check ensures the user is aware of the deletion action.

2. Update Job Status on Failure (AN060R)

•**Logic:** Updates the job status to indicate failure in the historical records.

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

•**Fields:** ahstl1_dato (Job date), ahstl1_ttid (Job type ID)

•**Condition:** This update occurs when the user selects option '7' to mark a job as failed.

–

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:** Called to display the populated subfile of historical records.

•**Logic:** This subroutine formats and presents the records in the user interface.

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

2. xd1win (Delete Confirmation Window)

•**Trigger:** Invoked when the user opts to delete a record.

•**Logic:** Displays a confirmation window for the deletion action.

•**Impact:** This ensures that the user confirms their intent to delete, preventing accidental deletions.

3. xdatsf (Date Formatting Subroutine)

•**Trigger:** Called to format date fields for display and processing.

•**Logic:** Converts date formats between internal and display formats.

•**Impact:** This is critical for maintaining accurate date representations throughout the program.