AN060R.MBR

Path: NXCLOUD/rpgsrc/AN060R.MBR **Generated**: 2026-01-08 12:34:25 **Processing Time**: 11787ms

Business Logic for Night Job History Query

This document outlines the business rules that govern the querying of night job history, based on an analysis of the RPG program AN060R. The primary focus is on the logic that processes user interactions with the subfile displaying historical records of night jobs.

The core logic for querying night job history is contained within the *inzsr subroutine in AN060R. The program interacts with files to display, update, and delete records related to night jobs, allowing users to maintain historical data efficiently.

Order Status and Header Rules

Night Job History: ahstpfr

1. Display Records

- **Logic:** The program retrieves and displays records from the ahstpfr file in a subfile format, allowing users to view historical data.
- **File:** ahstpfr (Night Job History)
- **Field:** ahdato
- **Condition:** Records are displayed if the subfile is populated and the user has not selected an exit option.

2. Delete Records

- **Logic:** Users can delete selected records from the night job history.
- **File:** ahstpfr (Night Job History)
- **Field:** b1valg
- **Condition:** The delete action is triggered when the user selects option '4' from the subfile.

Configuration and Authorization Rules

1. Update Night Job Status

- **Logic:** The program updates the status of a night job record to indicate it has failed.
- **Files:**
- ahstpfr (Night Job History)
- ahstlur (Night Job Update Log)
- **Fields:**
- ahstl1_dato (Date of the night job)
- ahstl1_ttid (Time ID of the night job)
- **Condition:** This logic is executed when the user selects option '7' to acknowledge a failed night job.

2. Clear Subfile

- **Logic:** The program clears the subfile before populating it with new records.
- **File:** b1sfl (Subfile for Night Job History)
- **Field:** srrn01

•**Condition:** This action occurs at the beginning of the program or when refreshing the display.

Financial and Transactional Rules

1. Insert New Records

•**Logic:** New records can be added to the night job history based on user input.

•**File:** ahstpfr (Night Job History)

•**Fields:**

•ahstl1_dato (Date of the night job)

•ahstl1_ttid (Time ID of the night job)

•**Condition:** The insertion occurs when the user provides valid data and confirms the action.

2. Check for Existing Records

•**Logic:** The program checks if a record already exists before performing updates.

•**File:** ahstpfr (Night Job History)

•**Condition:** This check is performed when the user attempts to update a record.

Special Conditions (Program-Specific)

1. Handle Subfile Navigation (AN060R)

•**Logic:** The program allows users to navigate through records in the subfile using function keys.

•**File:** b1sfl (Subfile for Night Job History)

•**Field:** srrn01

•**Condition:** Navigation occurs based on user input and the state of the subfile.

2. Display Confirmation Window (AN060R)

•**Logic:** A confirmation window is displayed when the user attempts to delete a record.

•**File:** d1win (Confirmation Window)

•**Fields:** d1dato (Date to confirm deletion)

•**Condition:** This window appears when the delete option is selected.

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 night job history.

•**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 the night job history.

2. xdatsf (Convert Screen to File Date)

•**Trigger:** Called when converting date formats between the screen and the file.

•**Logic:** This subroutine handles the transformation of date fields for consistency.

•**Impact:** Ensures that date information is accurately represented in both the user interface and the database.

3. xd1win (Delete Confirmation Window)

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

•**Logic:** Displays a confirmation window to prevent accidental deletions.

- **Impact:** This represents a **critical control point** in the process to safeguard data integrity.