RX000R.MBR

Path: NXCLOUD/rpgsrc/RX000R.MBR **Generated**: 2026-01-09 10:13:45 **Processing Time**: 12115ms

Business Logic for RX000R

This document outlines the business rules that govern the retrieval of file and path information for FTP jobs, based on an analysis of the RPG program RX000R. The primary focus is on how the program processes records to determine the appropriate file and path for FTP operations.

The core logic for retrieving file and path information is contained within the main processing section of the RX000R program. The program checks for the existence of records in specified files and handles conditions where certain data may be missing.

Order Status and Header Rules

RX000R: FrscdL1, FaftpL1

1. Record Existence Check

- **Logic:** The program checks if a record exists in the FrscdL1 file using the STSKEY chain operation.
- **File:** FrscdL1 (File containing job-related information)
- **Field:** STSFIR
- **Condition:** The process will not proceed if the record is not found in FrscdL1.

2. Local Path Validation

- **Logic:** If the record is found, it retrieves the firm name and checks for the local path in the FaftpL1 file.
- **File:** FaftpL1 (File containing FTP parameters)
- **Field:** aflocd
- **Condition:** The process will generate an error message if the local path (aflocd) is empty or if the record is not found in FaftpL1.

Configuration and Authorization Rules

1. Firm Code Conversion

- **Logic:** The program converts the firm code from a packed decimal format to a character format for further processing.
- **Files:**
- FrscdL1 (File containing job-related information)
- FaftpL1 (File containing FTP parameters)
- **Fields:**
- stsfir (from FrscdL1)
- aftpl1_firm (from FaftpL1)
- **Condition:** The conversion occurs every time the program runs, ensuring that the firm code is always in the correct format.

2. Error Handling for Missing Local Path

- **Logic:** If the local path is missing, an error message is generated and passed to a subroutine for handling.

•**File:** FaftpL1 (File containing FTP parameters)

•**Field:** aflocd

•**Condition:** The error message is triggered if aflocd is empty or if the record is not found.

Financial and Transactional Rules

1. Job Name Assignment

•**Logic:** The program assigns the job name from the record to a variable for further processing.

•**File:** FaftpL1 (File containing FTP parameters)

•**Fields:**

•p_job (assigned job name)

•rcjnav (job name from the record)

•**Condition:** The job name is assigned only if the record is found and valid.

2. File Name Assignment

•**Logic:** The program retrieves the file name from the record and assigns it to a variable.

•**File:** FaftpL1 (File containing FTP parameters)

•**Condition:** The file name is assigned only if the record is found and valid.

Special Conditions (Program-Specific)

1. Initialization Subroutine (RX000R)

•**Logic:** The program initializes parameters for file and path processing upon entry.

•**File:** N/A (Initialization logic)

•**Field:** N/A

•**Condition:** The initialization occurs every time the program is called, ensuring that parameters are set correctly.

2. Key List Definition (RX000R)

•**Logic:** The program defines key fields for file access to ensure proper record retrieval.

•**File:** FrscdL1, FaftpL1 (Files containing job-related information and FTP parameters)

•**Fields:** STSFIR (from FrscdL1), aftpl1_firm, aftpl1_navn (from FaftpL1)

•**Condition:** Keys are defined at the start of the program to facilitate efficient data access.

Subprogram Calls Affecting Logic

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

1. AA005R (Error Handling Subroutine)

•**Trigger:** Called when the local path is missing.

•**Logic:** This subroutine handles error messages related to missing local paths.

•**Impact:** This call acts as a **major logical gateway** for error reporting in the FTP job processing.

2. *INZSR (Initialization Subroutine)

•**Trigger:** Automatically called at the start of the program.

•**Logic:** Initializes parameters for the program.

•**Impact:** Ensures that all necessary parameters are set before processing begins.

3. KLIST (Key List Definition)

- **Trigger:** Defined at the beginning of the program.
- **Logic:** Establishes key fields for accessing records in FrscdL1 and FaftpL1.
- **Impact:** This represents the foundation for data retrieval logic in the program, ensuring efficient access to necessary records.