

RX000R.MBR

Path: NXCLOUD/rpgsrc/RX000R.MBR **Generated:** 2026-01-08 15:43:00 **Processing Time:** 19277ms

Business Logic for RX000R

This document outlines the business rules that govern the retrieval of file and path information for an FTP job, based on an analysis of the RPG program RX000R. The primary focus is on the logic that processes and validates file paths and firm identifiers.

The core logic for retrieving the necessary information is contained within the main processing section of the program. The program checks for the existence of records in specific files and assigns values to parameters based on the results of those checks.

Order Status and Header Rules

RX000R: rscdL1, aftpL1

1. Record Existence Check

- **Logic:** The program checks if a record exists in the rscdL1 file based on the STSKEY. If found, it retrieves the firm identifier and name.
- **File:** rscdL1 (Resource Code File)
- **Field:** STSFIR
- **Condition:** The process will not proceed if the record is not found in rscdL1.

2. FTP Parameter Validation

- **Logic:** If the record in aftpL1 is not found or if the location code (aflocd) is empty, an error message is generated indicating that the local path is missing.
- **File:** aftpL1 (FTP Parameters File)
- **Field:** aftpL1_firm, aftpL1_navn
- **Condition:** The process will generate an error if either the record is not found or the location code is blank.

Configuration and Authorization Rules

1. Firm Identifier Assignment

- **Logic:** The firm identifier is converted from a decimal representation and assigned to aftpL1_firm for further processing.

• **Files:**

- rscdL1 (Resource Code File)
- aftpL1 (FTP Parameters File)

• **Fields:**

- stsfir (from rscdL1)
- aftpL1_firm (from aftpL1)

- **Condition:** The assignment occurs only if the record in rscdL1 is found.

2. Path and File Assignment

- **Logic:** If the record is found in aftpL1, the program assigns the local path, file name, and job name to their respective variables.

- **File:** aftpL1 (FTP Parameters File)
- **Fields:** p_bane, p_file, p_job
- **Condition:** This assignment occurs only if the record in aftpL1 is found.

Financial and Transactional Rules

1. Error Handling for Missing Path

- **Logic:** If the local path is missing, an error message is generated and a specific subroutine is called to handle this error.

- **File:** aftpL1 (FTP Parameters File)

- **Fields:**

- w_m50 (Error Message)

- **Condition:** This logic is triggered if either the record is not found in aftpL1 or if aflocd is empty.

2. Program Termination

- **Logic:** The program sets the last record indicator to on and returns control to the calling program.

- **File:** None

- **Condition:** This occurs at the end of the program execution.

Special Conditions (Program-Specific)

1. Initialization Subroutine (RX000R)

- **Logic:** The initialization subroutine sets up the parameters for the program and defines the key fields for file access.

- **File:** rscdL1, aftpL1 (Resource Code File, FTP Parameters File)

- **Field:** STSFIR, aftpL1_firm, aftpL1_navn

- **Condition:** This subroutine is invoked at the start of the program to prepare for processing.

2. Key Field Definitions (RX000R)

- **Logic:** The program defines key fields for accessing the rscdL1 and aftpL1 files, ensuring that the correct records are accessed during processing.

- **File:** rscdL1, aftpL1 (Resource Code File, FTP Parameters File)

- **Fields:** STSFIR, aftpL1_firm, aftpL1_navn

- **Condition:** This definition is crucial for the correct functioning of the record retrieval logic.

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:** This program is called when the local path is missing.

- **Logic:** It handles the error by displaying the error message stored in w_m50.

- **Impact:** This call acts as a **major logical gateway** for managing errors related to missing FTP parameters.

2. *INZSR (Initialization Subroutine)

- **Trigger:** This subroutine is called at the beginning of the program.

- **Logic:** It initializes the parameters and sets up the key fields for file access.

- **Impact:** This is a **critical setup step** that ensures the program has the necessary context for execution.

3. CHAIN Operations (Record Retrieval)

- **Trigger:** The program uses the CHAIN operation to retrieve records from rscdL1 and aftpL1.
- **Logic:** It checks for the existence of records based on keys and assigns values accordingly.
- **Impact:** This represents the **core data retrieval logic** essential for the program's functionality.