

# RX000R.MBR

**Path:** NXCLOUD/rpgsrc/RX000R.MBR **Generated:** 2026-01-08 15:04:26 **Processing Time:** 17464ms

## 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 how the program processes records to obtain necessary parameters for FTP operations.

The core logic for retrieving file and path information is contained within the main processing section of the RX000R program. It reads from specific files and checks conditions to ensure that the necessary data is available for further processing.

### Order Status and Header Rules

RX000R: FrscdL1, FaftpL1

#### 1. Retrieve Firm Information

- **Logic:** The program retrieves the firm information based on the p\_firm parameter and checks if the corresponding record exists in the rscdL1 file.
- **File:** FrscdL1 (Firm Code Register)
- **Field:** l\_firm
- **Condition:** The process will not proceed if the firm record is not found in rscdL1.

#### 2. Check for Local Path

- **Logic:** If the firm record is found, it checks the aftpl1 file for the local path. If the local path is missing or not found, an error message is generated.
- **File:** FaftpL1 (FTP Parameters)
- **Field:** aflocl
- **Condition:** The process will generate an error if aflocl is blank or if the record in aftpl1 is not found.

### Configuration and Authorization Rules

#### 1. Parameter Initialization

- **Logic:** The program initializes parameters for file, path, firm, and job based on the input parameters received.

##### • **Files:**

- FrscdL1 (Firm Code Register)
- FaftpL1 (FTP Parameters)

##### • **Fields:**

- p\_firm (input firm code)
- p\_bane (output local path)

- **Condition:** The program must receive valid parameters for successful initialization.

#### 2. Error Handling for Missing Local Path

- **Logic:** If the local path is missing, an error message is prepared and passed to the error handling program.

- **File:** FaftpL1 (FTP Parameters)
- **Field:** aflocl
- **Condition:** The error handling is triggered if aflocl is blank.

## Financial and Transactional Rules

### 1. File and Job Assignment

- **Logic:** Upon successful retrieval of the local path, the program assigns the file name and job name to the respective output parameters.

- **File:** FaftpL1 (FTP Parameters)

#### • **Fields:**

- p\_file (output file name)

- p\_job (output job name)

- **Condition:** This assignment occurs only if the local path is successfully retrieved.

### 2. Completion of Process

- **Logic:** The program sets the last record indicator to indicate completion and prepares to return control to the calling program.

- **File:** None (finalization step)

- **Condition:** This step is executed unconditionally at the end of processing.

## Special Conditions (Program-Specific)

### 1. Initialization Subroutine (RX000R)

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

- **File:** FrscdL1 (Firm Code Register)

- **Field:** STSFIR

- **Condition:** This subroutine is called at the beginning of the program to prepare for data retrieval.

### 2. Key List Definition (RX000R)

- **Logic:** The program defines key lists for accessing the rscdL1 and aftpL1 files based on the firm and name.

- **File:** FaftpL1 (FTP Parameters)

- **Fields:** aftpL1\_firm (firm key), aftpL1\_navn (name key)

- **Condition:** This key list definition is essential for subsequent data retrieval operations.

## 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 Program)

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

- **Logic:** It receives an error message indicating that the local path is missing.

- **Impact:** This call acts as a **major logical gateway** for error handling, ensuring that missing data is reported appropriately.

### 2. \*INZSR (Initialization Subroutine)

- Trigger:** This subroutine is called at the start of the program.
- Logic:** It initializes the parameters and sets up key fields for file access.
- Impact:** This is a **critical setup step** that ensures the program has the necessary context to operate correctly.

### 3. CHAIN Operations

- Trigger:** The program performs CHAIN operations to retrieve records from rscdL1 and aftpl1.
- Logic:** It checks for the existence of records based on the keys defined.
- Impact:** These operations are essential for **data retrieval**, impacting the flow of the program based on the availability of records.