AX003R.MBR

Path: NXCLOUD/rpgsrc/AX003R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 11881ms

Business Logic for AX003R

This document outlines the business rules that govern the processing of financial data and XML output generation, based on an analysis of the RPG program AX003R. The primary focus is on the logic that handles the reading of financial records and the generation of XML formatted output.
The core logic for processing financial records is contained within the StartP subroutine in AX003R . This subroutine reads financial records from the rhtrl5 file and generates XML output based on the data retrieved.

Order Status and Header Rules

AX003R: rhtrl5, rxmll1

1. Firm Selection

•**Logic:** The program sets the firm number to be processed based on the user input.
•**File:** rhtrl5 (Financial records)
•**Field:** rhtrl5_firm
•**Condition:** The firm number is set to the user-defined firm number (w_firmn).

2. Record Filtering by Date

•**Logic:** The program filters records based on the specified date range.
•**File:** rhtrl5 (Financial records)
•**Field:** rhtrl5_bdat
•**Condition:** The record is processed if the record date (rbraar and rbrper) falls within the specified periods (p_pta and p_ptm).

Configuration and Authorization Rules

1. User Authorization

•**Logic:** The program checks if the user has the necessary permissions to access the financial records.
•**Files:**
•rhtrl5 (Financial records)
•xmlut (XML output)
•**Fields:**
•l_user (User identifier from the input)
•**Condition:** The program processes records only if the user is authorized based on the firm number.

2. Firm Initialization

•**Logic:** The program initializes the firm number for processing.
•**File:** rhtrl5 (Financial records)
•**Field:** rhtrl5_firm
•**Condition:** The firm number is initialized to the user-defined firm number at the start of the processing.

Financial and Transactional Rules

1. Record Processing Loop

•**Logic:** The program processes each financial record in a loop until all relevant records are read.

•**File:** rhtrl5 (Financial records)

•**Fields:**

•rbbiln (Bill number)

•rbbdat (Bill date)

•**Condition:** The loop continues while not at the end of the file and the bill number matches the specified criteria.

2. XML Output Generation

•**Logic:** The program generates XML output for each processed financial record.

•**File:** rxmll1 (XML output)

•**Condition:** The XML output is generated based on the financial record data until the end of the file is reached.

Special Conditions (Program-Specific)

1. Debugging Conditions (AX003R)

•**Logic:** The program includes debugging logic to check specific conditions for financial records.

•**File:** rhtrl5 (Financial records)

•**Field:** rbbiln

•**Condition:** If the bill number is 704 and account number is 7740, the program allows for additional checks. Note: This is primarily for debugging purposes.

2. Handling Specific XML Tags (AX003R)

•**Logic:** The program handles specific XML tags based on the type of data being processed.

•**File:** rxmll1 (XML output)

•**Fields:** axxmlt (XML template), text (Text to be included in XML)

•**Condition:** The program checks for specific conditions to determine how to format the XML output based on the financial data.

Subprogram Calls Affecting Logic

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

1. sxmlut (XML Output Subroutine)

•**Trigger:** Called during the processing of each financial record.

•**Logic:** This subroutine formats and writes the XML output based on the current financial record.

•**Impact:** This call acts as a **major logical gateway** for generating the XML output.

2. BehKonto (Account Processing Subroutine)

•**Trigger:** Called for each account associated with a financial record.

•**Logic:** This subroutine processes all lines related to a specific account and generates corresponding XML output.

•**Impact:** This ensures that all account-related data is accurately represented in the XML output.

3. redefnum (Number Formatting Subroutine)

- **Trigger:** Called to format numeric fields before they are included in the XML output.
- **Logic:** This subroutine formats numbers to ensure they meet the required specifications for XML.
- **Impact:** This is a **critical formatting step** that ensures numeric data is correctly represented in the output.