AX003R.MBR

Path: NXCLOUD/rpgsrc/AX003R.MBR **Generated**: 2026-01-08 15:42:59 **Processing Time**: 17646ms

Business Logic for XML Generation from Financial Records

This document outlines the business rules that govern the generation of XML output from financial records, based on an analysis of the RPG program AX003R. The primary focus is on how financial data is processed and converted into XML format for reporting purposes.

The core logic for XML generation is contained within the SXMLUT subroutine in AX003R. The program reads financial records from the RHTRPF file and generates XML output based on specific conditions and data fields.

Order Status and Header Rules

XML Generation Process: RHTRPF, AXMLUT

1. Initialize Financial Record

- **Logic:** The program initializes the financial record by setting the firm, year, period, and other relevant fields to prepare for reading records.
- **File:** RHTRPF (Financial Record File)
- **Field:** rhtrl5_firm, rhtrl5_raar, rhtrl5_rper, rhtrl5_biln, rhtrl5_bdat
- **Condition:** The process will not select a record if rbbiln is less than the specified p_frabil.

2. Read Financial Records

- **Logic:** The program reads through the financial records until it reaches the end of the file or meets specific conditions.
- **File:** RHTRPF (Financial Record File)
- **Field:** rbfirm, rbraar, rbrper
- **Condition:** The program continues reading records as long as the firm matches w_firmn and the year and period are within the specified limits.

Configuration and Authorization Rules

1. Set XML Output Header

- **Logic:** The program sets the XML output header to 'HOVEDBOK' to indicate the start of the financial report.
- **Files:**
- AXMLUT (XML Output File)
- **Fields:**
- rxmll1_head (XML Header)
- **Condition:** This action occurs at the beginning of the processing loop for financial records.

2. Skip Processing Based on Conditions

- **Logic:** The program checks for specific conditions that may require skipping certain records or processing steps.
- **File:** RHTRPF (Financial Record File)

- **Field:** axmaop
- **Condition:** If axmaop is equal to 'S', the program will skip to the next iteration without processing the current record.

Financial and Transactional Rules

1. Generate XML for Financial Transactions

- **Logic:** The program generates XML entries for each financial transaction, including details like account number, transaction date, and amounts.
- **File:** RHTRPF (Financial Record File)
- **Fields:**
- rbkont (Account Number)
- rbbdat (Transaction Date)
- **Condition:** XML entries are created for transactions that match the current financial record being processed.

2. Handle Currency Information

- **Logic:** The program checks for currency-related fields and generates XML entries accordingly.
- **File:** RHTRPF (Financial Record File)
- **Condition:** If the currency code is not 'NOK' and the amount is not zero, the program generates XML entries for currency details.

Special Conditions (Program-Specific)

1. Debugging and Testing Logic (AX003R)

- **Logic:** The program contains specific debugging logic that allows for testing of certain financial records based on predefined conditions.
- **File:** RHTRPF (Financial Record File)
- **Field:** rbbiln, rbkont, rbrper
- **Condition:** Debugging logic is triggered when specific values are encountered, allowing for manual inspection of records.

2. Finalizing XML Output for Each Record (AX003R)

- **Logic:** The program finalizes the XML output for each financial record, ensuring all necessary data is included before moving to the next record.
- **File:** AXMLUT (XML Output File)
- **Fields:** xmlrec (XML Record)
- **Condition:** The finalization occurs after processing all relevant fields for a given financial record.

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 Generation)

- **Trigger:** Called during the processing of each financial record.
- **Logic:** This subroutine handles the actual generation of XML entries based on the current financial record's data.

- **Impact:** This call acts as a **major logical gateway** for transforming financial data into XML format.

2. BehKonto (Account Processing)

- **Trigger:** Called for each account associated with the current financial record.
- **Logic:** This subroutine processes all relevant lines for the current account and generates corresponding XML entries.
- **Impact:** This is a **critical step** in ensuring all account-related data is accurately represented in the XML output.

3. redefnum (Number Formatting)

- **Trigger:** Called when formatting numeric fields for XML output.
- **Logic:** This subroutine ensures that numeric values are formatted correctly for XML representation.
- **Impact:** This ensures that all numeric data is presented in a consistent and expected format, enhancing the integrity of the XML output.