AX003R.MBR

Path: NXCLOUD/rpgsrc/AX003R.MBR **Generated**: 2026-01-08 12:04:38 **Processing Time**: 12945ms

Business Logic for XML Generation from Financial Records

This document outlines the business rules that govern the XML generation process from financial records, based on an analysis of the RPG programs AX003R. The primary focus is on the logic that handles reading financial records and formatting them into XML output.

The core logic for generating XML output from financial records is contained within the SXMLUT subroutine in AX003R. The program processes records from the RHTRL5 and RXMLL1 files, generating XML formatted data based on specific conditions and rules.

Order Status and Header Rules

Process_Name: AX003R: RHTRL5, RXMLL1

1. Firm Selection

- **Logic:** The program initializes the firm to be processed based on input parameters.
- **File:** RHTRL5 (Financial Records)
- **Field:** rhtrl5_firm
- **Condition:** The firm is set to the value of w_firmn at the start of processing.

2. Record Filtering

- **Logic:** The program reads records from RHTRL5 until it reaches the end of the file or until the record's year and period do not match the specified parameters.
- **File:** RHTRL5 (Financial Records)
- **Field:** rbraar, rbrper
- **Condition:** The process continues while rbfirm = w_firmn and either rbraar < p_pta or rbraar = p_pta and rbrper <= p_ptm.

Configuration and Authorization Rules

1. XML Header Initialization

- **Logic:** The program sets the XML header for the output based on the current processing context.
- **Files:**
- RXMLL1 (XML Lines)
- **Fields:**
- rxmll1_head (XML Header)
- s_head (Current Header)
- **Condition:** The header is initialized with the value 'HOVEDBOK' at the start of processing.

2. End of Record Processing

- **Logic:** The program writes any remaining XML lines for the last processed record.
- **File:** RXMLL1 (XML Lines)
- **Field:** axhead
- **Condition:** The loop continues until the end of the RXMLL1 file is reached and the header matches 'HOVEDBOK'.

Financial and Transactional Rules

1. Account Processing

- **Logic:** For each financial record, the program processes account details and generates corresponding XML lines.
- **File:** RHTRL5 (Financial Records)
- **Fields:**
- rbkont (Account Number)
- rbbiln (Bill Number)
- **Condition:** The loop continues while rbbiln = s_biln and rbbdat = s_bdat.

2. Currency Handling

- **Logic:** The program checks for currency values and generates XML entries for currency codes and amounts.
- **File:** RHTRL5 (Financial Records)
- **Condition:** If rbvalk is not blank or 'NOK' and rbvalb is not zero and does not equal rbneto, XML entries for currency code and amount are generated.

Special Conditions (Program-Specific)

1. Debugging Logic (AX003R)

- **Logic:** The program includes debugging conditions to verify specific account numbers and periods during processing.
- **File:** RHTRL5 (Financial Records)
- **Field:** rbbiln, rbkont, rbrper
- **Condition:** If rbbiln = 704, rbkont = 7740, and rbrper = 09, the program evaluates rbrper without alteration. Note: This is a debugging feature and should be removed in production.

2. XML Line Generation (AX003R)

- **Logic:** The program generates XML lines based on various conditions, including handling special characters and formatting.
- **File:** RXMLL1 (XML Lines)
- **Fields:** axxmlt, text
- **Condition:** The program checks for specific markers in the XML template and replaces them with appropriate values or skips them based on the defined logic.

Subprogram Calls Affecting Logic

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

1. SXMLUT (XML Line Output)

- **Trigger:** This subroutine is called multiple times to generate XML lines based on the current processing context.
- **Logic:** It formats the output based on the current record's values and specific XML rules.
- **Impact:** This call acts as a **major logical gateway** for generating the XML output.

2. BehKonto (Account Processing)

- **Trigger:** This subroutine is called for each account to process its details and generate XML lines.
- **Logic:** It handles the reading of account-specific records and formats them into XML.
- **Impact:** This is a **critical step** in ensuring all account details are accurately represented in the XML output.

3. redefnum (Number Formatting)

- **Trigger:** This subroutine is called to format numeric fields for XML output.
- **Logic:** It ensures that numbers are formatted correctly, handling leading zeros and decimal points.
- **Impact:** This ensures that numeric values in the XML are correctly formatted for downstream processing.