AX003R.MBR

Path: NXCLOUD/rpgsrc/AX003R.MBR **Generated**: 2026-01-08 13:32:03 **Processing Time**: 14279ms

Business Logic for XML Generation from Financial Records

This document outlines the business rules that govern the process of generating XML output from financial records, 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 corresponding XML output.

The core logic for processing financial records and generating XML is contained within the SXMLUT subroutine in AX003R. The program reads data from the RHTRL5 and RXMLL1 files, processes the records based on various conditions, and outputs the results in XML format.

Order Status and Header Rules

Process_Name: AX003R: RHTRL5, RXMLL1

1. Firm Selection

•**Logic:** The program initializes the firm selection based on the input parameters and sets the key for reading records.

•**File:** RHTRL5 (Financial Records)

•**Field:** rhtrl5_firm

•**Condition:** The process will not select records if the firm number does not match the input firm number (w_firmn).

2. Date Range Filtering

•**Logic:** The program filters records based on the accounting period and year.

•**File:** RHTRL5 (Financial Records)

•**Fields:** rhtrl5_raar, rhtrl5_rper

•**Condition:** Records are selected if the year is less than the end year (p_pta) or if it matches the end year but the period is less than or equal to the end period (p_ptm).

Configuration and Authorization Rules

1. XML Output Initialization

•**Logic:** The program initializes the XML output structure before processing records.

•**Files:**

•RXMLL1 (XML Template)

•**Fields:**

•rxmll1_head (XML Header)

•rxmll1_linj (XML Line)

•**Condition:** The XML output is generated based on the financial records being processed.

2. Record Processing Control

•**Logic:** The program controls the processing of financial records based on the current record's state.

•**File:** RHTRL5 (Financial Records)

•**Field:** rbbiln

•**Condition:** The process continues to read records until the end of the file is reached or the specified conditions are met.

Financial and Transactional Rules

1. Transaction Line Processing

•**Logic:** Each transaction line is processed to generate corresponding XML entries.

•**File:** RHTRL5 (Financial Records)

•**Fields:**

•rbbiln (Bill Number)

•rbbdat (Bill Date)

•**Condition:** The program processes lines as long as they match the current bill number and date.

2. Currency Handling

•**Logic:** The program checks for currency discrepancies and formats the output accordingly.

•**File:** RHTRL5 (Financial Records)

•**Condition:** If the currency values are not standard, the program adjusts the XML output to reflect the currency code and amount.

Special Conditions (Program-Specific)

1. Error Handling in XML Generation (AX003R)

•**Logic:** The program includes error handling for XML generation, skipping lines that do not meet criteria.

•**File:** RXMLL1 (XML Template)

•**Field:** axmaop

•**Condition:** Lines marked for skipping (e.g., axmaop = 'S') are not included in the output. Note: This check is crucial for ensuring only valid records are processed.

2. Finalization of XML Output (AX003R)

•**Logic:** The program finalizes the XML output by writing any remaining lines after processing all records.

•**File:** RXMLL1 (XML Template)

•**Fields:** axlinj (Line Number)

•**Condition:** The program writes the final XML entries until the end of the file is reached.

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

•**Logic:** This subroutine generates the XML output for each line based on the current 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 a transaction.

•**Logic:** This subroutine processes all lines related to a specific 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 to format numerical values before outputting to XML.

•**Logic:** This subroutine ensures that numbers are formatted correctly, handling leading zeros and
 decimal points.

•**Impact:** This ensures that numerical data is presented in a **consistent and user-friendly
 manner** in the final XML output.