

AP050R.MBR

Path: NXCLOUD/rpgsrc/AP050R.MBR **Generated:** 2026-01-08 12:16:23 **Processing Time:** 11135ms

Business Logic for Profile Manager API Interaction

This document outlines the business rules that govern the interaction with the Profile Manager API, based on an analysis of the RPG program AP050R. The primary focus is on the logic for sending requests to the API and processing the responses.

The core logic for handling the API requests and responses is contained within the main program structure of AP050R. The program builds requests, interacts with the API, and processes the returned data while managing temporary files on the Integrated File System (IFS).

API Request and Response Handling Rules

Profile Manager API Interaction: AP050R

1. Request File Creation

- Logic:** The program creates a request file to send to the Profile Manager API, including a token in the request header.
- File:** IFS_Output1 (Request file for API)
- Field:** w_numm
- Condition:** The request file is created only if the directory change to the specified path is successful.

2. Header File Creation

- Logic:** The program creates a headers file that includes the API key for authentication.
- File:** IFS_Output2 (Headers file for API)
- Field:** w_token
- Condition:** The headers file is created only if the directory change to the specified path is successful.

3. API Call Execution

- Logic:** The program constructs the API call with the necessary parameters and executes it using a separate program (AW702C).
- File:** API_Input (Response file)
- Field:** w_url2, w_reqt, w_reqf, w_rspf
- Condition:** The API call is made only if the request file and headers file are successfully created.

4. Response File Check

- Logic:** The program checks if the response file from the API is available after the call.
- File:** API_Input (Response file)
- Condition:** If the response file cannot be opened, the API status is set to '9', indicating an error.

Error Handling and Cleanup Rules

1. Error Handling

- Logic:** If the API call fails, the program sets the status to '8' and proceeds to cleanup.

- File:** N/A
- Condition:** Triggered by an unsuccessful API call.

2. File Cleanup

- Logic:** The program attempts to delete temporary files created during the process.
 - Files:**
 - IFS_Output1 (Request file)
 - IFS_Output2 (Headers file)
 - IFS_Input (Response file)
 - Condition:** Cleanup occurs regardless of the API call's success or failure.
-

Initialization and Parameter Handling Rules

1. Parameter Initialization

- Logic:** The program initializes parameters and retrieves necessary values from the database for the API call.
- File:** afpspf, aposextnst (Database tables)
- Fields:**
 - w_url (API endpoint)
 - w_token (API key)
- Condition:** Parameters are set during the program's initialization phase.

2. Sequential Number Generation

- Logic:** The program retrieves a sequential number to be used in the API request.
 - File:** N/A
 - Condition:** This is done by calling another program (AS100R) to generate the number based on firm and other parameters.
-

Subprogram Calls Affecting Logic

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

1. AW702C (API Call Execution)

- Trigger:** Called to execute the API request after preparing the necessary parameters.
- Logic:** This program handles the actual HTTP request to the Profile Manager API.
- Impact:** This call is crucial as it represents the main interaction with the API, determining whether data is successfully retrieved.

2. AP051R (Response Processing)

- Trigger:** Called to process the response file if the API call is successful.
- Logic:** This program handles the extraction of data from the API response.
- Impact:** This step is essential for utilizing the data returned from the API, affecting subsequent business logic.

3. AS100R (Sequential Number Generation)

- Trigger:** Called during the initialization to obtain a sequential number for the API request.
- Logic:** This program retrieves the next number in a sequence based on firm-specific parameters.

•**Impact:** This is important for ensuring unique identifiers in API requests, which is critical for data integrity.

–

This documentation captures the essential business logic and rules governing the interaction with the Profile Manager API as implemented in the AP050R RPG program.