AP050R.MBR

Path: NXCLOUD/rpgsrc/AP050R.MBR **Generated**: 2026-01-08 12:26:04 **Processing Time**: 11730ms

Business Logic for Profile Manager API Lookup

This document outlines the business rules that govern the process of interacting with the Profile Manager API to retrieve user profile information, based on an analysis of the RPG program AP050R. The primary focus is on the logic for preparing and sending requests to the API, handling responses, and managing file operations.

The core logic for the Profile Manager API lookup is contained within the main program logic of AP050R. The program constructs requests, interacts with the API, and processes responses while managing temporary files on the Integrated File System (IFS).

API Request and Response Management Rules

ProfileManagerAPI: IFS_Output, IFS_Input

1. Request File Creation

- **Logic:** The program creates a request file on the IFS to send to the Profile Manager API. The filename is dynamically generated based on a numeric identifier.
- **File:** IFS_Output1 (Request file for API)
- **Field:** w_numm
- **Condition:** The request file is created if the directory change to the specified path is successful.

2. Headers File Creation

- **Logic:** A headers file is created to store the API key required for authentication. The program writes the API key into this file.
- **File:** IFS_Output2 (Headers file for API)
- **Field:** w_token
- **Condition:** The headers file is created if the directory change to the specified path is successful.

3. API Call Execution

- **Logic:** The program constructs the API call parameters and invokes an external program to perform the HTTP request to the Profile Manager API.
- **File:** w_url2 (API endpoint)
- **Field:** w_reqf, w_rspf
- **Condition:** The API call is executed with the constructed parameters, and the status is tracked.

4. Response File Handling

- **Logic:** The program checks for the existence of the response file from the API and processes it if available.
- **File:** IFS_Input (Response file from API)
- **Field:** API_status
- **Condition:** The response file is checked, and if it exists, the program proceeds to handle the data.

Error Handling and Cleanup Rules

1. Error Handling

- **Logic:** If the API call fails, the program sets an error status and proceeds to cleanup operations.
- **File:** N/A
- **Field:** p_stat
- **Condition:** The error routine is triggered if the API call does not succeed.

2. File Cleanup

- **Logic:** Temporary files created during the API interaction are deleted to maintain a clean environment.
- **File:** IFS_Output1, IFS_Output2, IFS_Input
- **Condition:** Cleanup operations are executed if certain conditions are met (currently commented out).

Initialization and Parameter Handling Rules

1. Parameter Initialization

- **Logic:** The program initializes parameters required for the API call, including firm and member identifiers.
- **File:** N/A
- **Field:** p_firm, p_memb
- **Condition:** Parameters are set during the initialization subroutine.

2. Dynamic URL Retrieval

- **Logic:** The program retrieves the API URL and authentication token from the database based on the firm identifier.
- **File:** afpspf, aposextnst
- **Fields:** w_url, w_token
- **Condition:** The URL and token are fetched using SQL queries based on the provided firm identifier.

Subprogram Calls Affecting Logic

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

1. AW702C (HTTP Request Handler)

- **Trigger:** Called to execute the HTTP request to the Profile Manager API.
- **Logic:** This program handles the actual sending of the request and receiving the response.
- **Impact:** This call acts as a **major logical gateway** for the API interaction.

2. AP051R (Response Processing)

- **Trigger:** Called to process the response file received from the API.
- **Logic:** This program extracts relevant data from the response file for further processing.
- **Impact:** This represents the handoff to the next major business function, ensuring that the data retrieved is utilized effectively.

3. AS100R (Sequence Number Generator)

- **Trigger:** Called to generate a unique sequence number for the API request.

- **Logic:** This program retrieves the next available sequence number based on the firm identifier.
- **Impact:** This ensures that each API request is uniquely identified, which is crucial for tracking and managing requests.

This document provides a comprehensive overview of the business logic implemented in the AP050R program, focusing on the interaction with the Profile Manager API and the associated file management processes.