AP050R.MBR

Path: NXCLOUD/rpgsrc/AP050R.MBR **Generated**: 2026-01-08 12:21:16 **Processing Time**: 10386ms

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 how the program constructs requests to the API and processes the responses.

The core logic for API interaction is contained within the main program logic of AP050R. The program prepares a request, sends it to the Profile Manager API, and handles the response, including file management on the Integrated File System (IFS).

API Request and Response Handling Rules

Profile Manager API Interaction: IFS_Output, IFS_Input

1. Request File Creation

- **Logic:** The program constructs a request file with a specific naming convention and creates it in the designated IFS path.
- **File:** IFS_Output1 (Request file for API)
- **Field:** w_numm
- **Condition:** The request file is created only if the directory change to the IFS path is successful.

2. Header File Creation

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

API Call and Response Processing Rules

1. API Call Execution

- **Logic:** The program constructs the necessary parameters and calls the external web service to fetch data from the Profile Manager API.
- **Files:**
- IFS_Output1 (Request file)
- IFS_Input (Response file)
- **Fields:**
- w_url2 (API endpoint)
- w_reqt (Request type)
- **Condition:** The API is called with the constructed parameters, and the response is expected to be written to the response file.

2. Response File Check

- **Logic:** After the API call, the program checks if the response file has been created successfully.
- **File:** IFS_Input (Response file)

- **Condition:** If the response file cannot be opened, the program sets the API status to indicate failure.

Cleanup and Error Handling Rules

1. File Cleanup

- **Logic:** The program attempts to delete temporary files created during the request process to maintain a clean IFS environment.
- **File:** IFS_Output1, IFS_Output2
- **Condition:** Cleanup occurs only if the program execution reaches the cleanup section without errors.

2. Error Handling

- **Logic:** If the API call fails, the program sets an error status and jumps to the cleanup section.
- **File:** N/A
- **Condition:** The error routine is triggered if the API call does not succeed, ensuring that the program exits gracefully.

Initialization and Parameter Handling Rules

1. Parameter Initialization

- **Logic:** The program initializes parameters for the API call, including firm and member identifiers.
- **File:** N/A
- **Condition:** Parameters are set based on input values when the program is invoked.

2. API Key Retrieval

- **Logic:** The program retrieves the API key from the COMPANY_SETTINGS table to authenticate requests.
- **File:** aposextnst (Settings table)
- **Field:** w_token
- **Condition:** The API key is fetched 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 (Web Service Call)

- **Trigger:** Called to execute the API request with the constructed parameters.
- **Logic:** This subprogram handles the actual HTTP request to the Profile Manager API.
- **Impact:** This call acts as a **major logical gateway** for data retrieval from the API.

2. AP051R (Response Processing)

- **Trigger:** Called after a successful API response to process the returned data.
- **Logic:** This subprogram handles the parsing and processing of the response file.
- **Impact:** This represents the handoff to the next major business function, ensuring that the data received from the API is utilized appropriately.

3. AS100R (Number Generation)

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

- **Logic:** This subprogram initializes a counter used in constructing the request file names.
- **Impact:** This is essential for ensuring that each request is uniquely identified, preventing file overwrites.

This documentation provides a comprehensive overview of the business logic encapsulated in the AP050R RPG program, detailing how it interacts with the Profile Manager API and manages file operations on the IFS.