AP050R.MBR

Path: NXCLOUD/rpgsrc/AP050R.MBR **Generated**: 2026-01-08 11:56:18 **Processing Time**: 10781ms

Business Logic for Profile Manager API Integration

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 API interaction is contained within the main program structure of AP050R. The program handles the creation of request files, sends requests to the Profile Manager API, and processes the responses received.

API Request and Response Handling Rules

Profile Manager API Integration: Profile Manager API

1. Create Request File

- **Logic:** The program creates a request file with a specific naming convention and a headers file that includes the API key.
- **File:** IFS_Output1 (Request file for API)
- **Field:** w_numm
- **Condition:** The request file is created in the directory specified by IFS_path if the directory change is successful.

2. Write API Key to Headers File

- **Logic:** The program writes the API key into the headers file for authentication purposes.
- **File:** IFS_Output2 (Headers file for API)
- **Field:** w_tokeut
- **Condition:** The headers file is opened successfully for writing before the API key is written.

API Call and Response Processing Rules

1. Send API Request

- **Logic:** The program constructs the API request and sends it using the AW702C subprogram, specifying the necessary parameters.
- **Files:**
- IFS_Output1 (Request file)
- IFS_Input (Response file)
- **Fields:**
- w_url2 (API endpoint)
- w_reqt (HTTP method)
- **Condition:** The API request is sent successfully, and the program checks for the existence of the response file.

2. Process API Response

- **Logic:** If the response file is received, the program calls another subprogram AP051R to handle the data extraction from the response.

- **File:** API_Input (Response file)
- **Condition:** The response file exists and is opened successfully for reading.

Error Handling and Cleanup Rules

1. Error Handling

- **Logic:** If the API call fails, the program sets an error status and proceeds to cleanup.
- **File:** N/A
- **Condition:** Triggered by a failure in the API call, indicated by checking the status of the API response.

2. Cleanup Temporary Files

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

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 Subprogram)

- **Trigger:** Called to send the constructed API request.
- **Logic:** This subprogram handles the HTTP request to the Profile Manager API, using parameters such as the URL, request type, and file paths.
- **Impact:** This call is critical for initiating the API interaction and determining the success of the request.

2. AP051R (Response Processing Subprogram)

- **Trigger:** Called to process the response received from the API.
- **Logic:** This subprogram extracts data from the API response file and updates the relevant business logic.
- **Impact:** Represents the handoff to the next major business function, ensuring that the data received is handled appropriately.

3. AS100R (Number Generation Subprogram)

- **Trigger:** Called to generate a unique number for the request.
- **Logic:** This subprogram retrieves a sequence number that is used in the naming of request files.
- **Impact:** Ensures that each request is uniquely identified, preventing conflicts in file handling.

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