AP050R.MBR

Path: NXCLOUD/rpgsrc/AP050R.MBR **Generated**: 2026-01-09 10:13:44 **Processing Time**: 10111ms

Business Logic for Profile Manager API Request

This document outlines the business rules that govern the process of making a request to the Profile Manager API to return user profile information, based on an analysis of the RPG program AP050R. The primary focus is on the logic surrounding API requests and responses.

The core logic for handling the API request is contained within the main program logic of AP050R. This program constructs a request to the Profile Manager API, sends it, and processes the response.

API Request and Response Handling Rules

ProfileManagerAPI: IFS Files

1. Create Request File

- **Logic:** The program creates a request file in the Integrated File System (IFS) to send to the Profile Manager API. The request file is initially empty, and the request header includes an authentication token.
- **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. Create Headers File

- **Logic:** A headers file is created in the IFS that contains the API key for authentication. This file is necessary for the API request to be accepted.
- **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.

API Call and Response Processing Rules

1. Make API Call

- **Logic:** The program constructs the necessary parameters and calls the AW702C subroutine to execute the API request. It specifies the URL, request method, and file paths for both the request and response.
- **Files:**
- IFS_Output1 (Request file)
- IFS_Input (Response file)
- **Fields:**
- w_url2 (API endpoint)
- w_reqt (HTTP method)
- w_reqf (Request file path)
- w_rspf (Response file path)
- **Condition:** The API call is made regardless of the current state, but the response is only processed if the API call is successful.

2. Check API Response

- **Logic:** The program checks if the response file is created successfully after the API call. If the file is not found, it sets an error status.
- **File:** IFS_Input (Response file)
- **Condition:** If the response file cannot be opened, the API_status is set to '9', indicating an error in receiving the response.

Cleanup and Error Handling Rules

1. Cleanup IFS Files

- **Logic:** The program attempts to delete the request and headers files from the IFS after processing the API response to clean up temporary files.
- **Files:**
- IFS_Output1 (Request file)
- IFS_Output2 (Headers file)
- IFS_Input (Response file)
- **Condition:** Cleanup occurs regardless of the success of the API call, but the actual deletion logic is commented out in the current version.

2. Error Handling

- **Logic:** If the API call fails, the program sets the status to '8' and jumps to the exit label to terminate the program gracefully.
- **Condition:** This error handling is triggered during the processing of the API response if an error is detected.

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 Subroutine)

- **Trigger:** Called to execute the API request after constructing the necessary parameters.
- **Logic:** This subroutine handles the actual HTTP request to the Profile Manager API using the provided parameters.
- **Impact:** This call acts as a **major logical gateway**, determining whether the API request is successful and whether a valid response is received.

2. AP051R (Response Processing Subroutine)

- **Trigger:** Called to process the response from the API if the response file is successfully created.
- **Logic:** This subroutine extracts relevant data from the API response file for further processing.
- **Impact:** This represents the handoff to the next major business function, where the response data is utilized.

3. AS100R (Number Generation Subroutine)

- **Trigger:** Called to initialize and retrieve a unique number for the API request.
- **Logic:** This subroutine generates a unique identifier for the request.
- **Impact:** This ensures that each API request is uniquely identified, which is crucial for tracking and managing requests.

This documentation provides a comprehensive overview of the business logic implemented in the AP050R program, focusing on the interaction with the Profile Manager API and the handling of request and response files.