

AP050R.MBR

Path: NXCLOUD/rpgsrc/AP050R.MBR **Generated:** 2026-01-08 12:06:04 **Processing Time:** 11952ms

Business Logic for Profile Manager API Integration

This document outlines the business rules that govern the process of interacting with the Profile Manager API, based on an analysis of the RPG program AP050R. The primary focus is on how the program constructs requests, handles responses, and manages files in the Integrated File System (IFS).

The core logic for the Profile Manager API integration is contained within the main program logic of AP050R. The program prepares a request to the Profile Manager API, sends the request, and processes the response received from the API.

API Request and Response Handling Rules

Profile Manager API Integration: IFS_Output, IFS_Input

1. Request File Creation

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

2. Header File Creation

- Logic:** A header file is created to store the API key required for authentication when making the API call.
- File:** IFS_Output2 (Header file for the API)
- Field:** w_token
- Condition:** The header file is created only if the directory change to the specified IFS path is successful.

API Call and Response Processing Rules

1. API Call Execution

- Logic:** The program constructs the necessary parameters and calls an external web service to send the request to the Profile Manager API.
- Files:**
 - IFS_Output1 (Request file)
 - IFS_Input (Response file)
- Fields:**
 - w_url (API endpoint)
 - w_reqt (HTTP method)
- Condition:** The API call is executed only if the request file and header file are successfully created.

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:** The program sets API_status to '9' if the response file cannot be opened, indicating an error in receiving the response.
-

Error Handling and Cleanup Rules

1. Error Handling

- Logic:** If the API call fails, the program sets an error status and jumps to the cleanup section.
- File:** N/A
- Condition:** Triggered by a failure in the API call or if the response file is not found.

2. File Cleanup

- Logic:** The program attempts to delete the request and header files created in the IFS after processing is complete.
 - Files:**
 - IFS_Output1 (Request file)
 - IFS_Output2 (Header file)
 - IFS_Input (Response file)
 - Condition:** Cleanup is attempted regardless of whether the API call was successful or not.
-

Initialization and Parameter Handling Rules

1. Parameter Initialization

- Logic:** The program initializes parameters and retrieves necessary values from the database, such as the API URL and token.
- Files:**
 - afpspf (File containing API URL)
 - aposextnst (File containing API token)
- Fields:**
 - w_url (API URL)
 - w_token (API token)
- Condition:** Parameters are initialized at the start of the program execution.

2. Member Processing

- Logic:** The program processes a member identifier to be used in the API request.
 - File:** N/A
 - Field:** p_memb
 - Condition:** The member identifier is passed as a parameter to the program.
-

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 after constructing the necessary parameters.

- Logic:** This program handles the actual HTTP request to the Profile Manager API.
- Impact:** This call acts as a **major logical gateway** for sending requests and receiving responses from the API.

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 received is handled appropriately.

3. AS100R (Number Generation)

- Trigger:** Called to generate a unique number used in the API request.
- Logic:** This program retrieves the next available number for the request.
- Impact:** This is crucial for ensuring that each API request is uniquely identified, preventing conflicts.

This documentation provides a comprehensive overview of the business logic implemented in the AP050R program, detailing how it interacts with the Profile Manager API and manages files within the IFS.