

# AP050R.MBR

**Path:** NXCLOUD/rpgsrc/AP050R.MBR **Generated:** 2026-01-08 12:57:59 **Processing Time:** 15677ms

## 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 the logic related to making requests to the API and handling responses.

The core logic for the Profile Manager API interaction is contained within the main program logic of AP050R. This program constructs requests to the Profile Manager API, processes the responses, and manages temporary files on the Integrated File System (IFS).

## API Request and Response Handling Rules

ProfileManagerAPI: AP050R

### 1. Request File Creation

- Logic:** The program creates a request file with a specific naming convention based on the member number. This file is initially empty and is used to send requests to the API.
- File:** IFS\_Output1 (Request file for API)
- Field:** w\_numm
- Condition:** The request file is created only if the directory change to the specified path is successful.

### 2. Header File Creation

- Logic:** A headers file is created to include the API token needed for authentication. This file is also named based on the member number.
- File:** IFS\_Output2 (Headers file for API)
- Field:** w\_token
- Condition:** The headers file is created only if the directory change to the specified path is successful.

### 3. API Call Execution

- Logic:** The program constructs the URL for the API call and sets parameters for the request type (GET) and file paths for the request and response. It then calls an external program AW702C to execute the API call.
- File:** N/A (External program call)
- Field:** w\_url2, w\_reqt, w\_reqf, w\_rspf
- Condition:** The API call is made regardless of the previous file operations, but it relies on the successful creation of the request and headers files.

### 4. Response File Handling

- Logic:** After the API call, the program checks if the response file has been created. If it exists, it processes the response by calling another program AP051R.
- File:** API\_Input (Response file)

- Field:** API\_status
  - Condition:** The response file must exist for the program to proceed with processing the response.
- 

## Error Handling and Cleanup Rules

### 1. Error Handling

- Logic:** If the API call fails (indicated by a status code), the program sets an error status and proceeds to cleanup.
- File:** N/A (Error handling logic)
- Field:** p\_stat
- Condition:** If the API call does not return a successful status, the program sets p\_stat to '8' and jumps to the cleanup section.

### 2. File Cleanup

- Logic:** The program attempts to delete temporary files created during the process, including request and response files.
  - File:** IFS\_Output1, IFS\_Output2, IFS\_Input
  - Condition:** Cleanup is attempted regardless of success or failure of the API call, but the actual deletion logic is commented out in the code.
- 

## Initialization and Parameter Handling Rules

### 1. Parameter Initialization

- Logic:** The program initializes parameters upon entry, including retrieving the API key and URL from the database based on the firm identifier.
- File:** afpspf, aposextnst (Database tables)
- Field:** w\_url, w\_token
- Condition:** These values are fetched from the database at the start of the program to ensure that the API call has the necessary credentials.

### 2. Member Number Generation

- Logic:** The program generates a unique member number for the API request by calling another program AS100R.
  - File:** N/A (External program call)
  - Field:** w\_numm
  - Condition:** The member number is generated based on the parameters provided and is essential for naming the request and response files.
- 

## 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 Execution)

- Trigger:** This program is called after constructing the request and headers files.
- Logic:** It executes the API call to the Profile Manager and handles the HTTP request.
- Impact:** This call is critical as it represents the actual interaction with the external API.

## **2. AP051R (Response Processing)**

- Trigger:** This program is called if the response file is successfully created.
- Logic:** It processes the data returned from the API call.
- Impact:** This call is essential for extracting and utilizing the information returned by the Profile Manager API.

## **3. AS100R (Member Number Generation)**

- Trigger:** This program is called during initialization to generate a unique member number.
- Logic:** It retrieves the next available member number for use in file naming.
- Impact:** This ensures that each API request is uniquely identified, which is important for tracking and managing requests.

–

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 related file operations.