

# AP050R.MBR

**Path:** NXCLOUD/rpgsrc/AP050R.MBR **Generated:** 2026-01-08 12:22:45 **Processing Time:** 11117ms

## Business Logic for Profile Manager API Lookup

This document outlines the business rules that govern the process of interacting with the Profile Manager API to retrieve user profile information, based on an analysis of the RPG program AP050R. The primary focus is on the logic for constructing requests, handling responses, and managing file operations related to the API interaction.

The core logic for the Profile Manager API lookup is contained within the main program logic of AP050R. The program constructs a request to the Profile Manager API, processes the response, and manages temporary files on the Integrated File System (IFS).

## Order Status and Header Rules

Profile Manager API Lookup: AP050R

### 1. Request Construction

- **Logic:** The program constructs a request file to send to the Profile Manager API, including a token in the headers.
- **File:** IFS\_Output1 (Request file for the API)
- **Field:** w\_token
- **Condition:** The request file is created if the directory change to the specified path is successful.

### 2. Response Handling

- **Logic:** After sending the request, the program checks for the existence of the response file to determine if the API call was successful.
- **File:** IFS\_Input (Response file from the API)
- **Field:** API\_status
- **Condition:** The response file must be successfully opened to proceed with data extraction.

## Configuration and Authorization Rules

### 1. Token Retrieval

- **Logic:** The program retrieves the API key (token) from the aposextnst table based on the provided firm parameter.
- **Files:**
  - aposextnst (Table containing API keys)
- **Fields:**
  - w\_token (API key)
- **Condition:** The token is fetched only if the firm matches the specified parameter and the system is set to 'EGAPPS'.

### 2. URL Configuration

- **Logic:** The program retrieves the base URL for the API from the afpspf table.
- **Files:**
  - afpspf (Table containing configuration settings)

**•Fields:**

•w\_url (Base URL for API requests)

**•Condition:** The URL is fetched based on the file identifier and a specific identifier for the cloud API.

## Financial and Transactional Rules

### 1. File Cleanup

**•Logic:** The program performs cleanup of temporary files created during the API interaction.

**•File:** IFS\_Output1, IFS\_Output2, IFS\_Input (Temporary files on IFS)

**•Condition:** Cleanup is executed regardless of the API call result, ensuring no residual files remain.

### 2. API Call Execution

**•Logic:** The program calls an external procedure to execute the API request and handle the response.

**•File:** AW702C (External procedure for web service call)

**•Condition:** The API call is made with the constructed request parameters, and the status is evaluated based on the response.

## Special Conditions (Program-Specific)

### 1. Error Handling (AP050R)

**•Logic:** If the web service call fails, the program sets an error status and proceeds to cleanup.

**•File:** N/A

**•Field:** p\_stat

**•Condition:** The program jumps to the error handling routine if the API call does not succeed.

### 2. Initialization Routine (AP050R)

**•Logic:** The program initializes parameters and retrieves necessary values from the database before processing.

**•File:** N/A

**•Fields:** p\_firm, p\_memb, p\_stat

**•Condition:** Initialization occurs at the entry point of the program to prepare for API interaction.

## 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 Procedure)

**•Trigger:** Called after constructing the request and before handling the response.

**•Logic:** This procedure executes the API call using the constructed parameters and manages the HTTP request.

**•Impact:** This call acts as a **major logical gateway** to the Profile Manager API, determining the success or failure of the interaction.

### 2. AP051R (Response Processing Procedure)

**•Trigger:** Called if the API response file is successfully opened.

- Logic:** This procedure processes the data returned from the API response file.
- Impact:** This represents the handoff to the next major business function, extracting relevant user profile information.

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

- Trigger:** Called during initialization to generate a unique number for the API request.
- Logic:** This procedure retrieves a sequential number to be used in constructing the request file names.
- Impact:** This ensures that each request is uniquely identified, preventing conflicts in file handling.