

# AP050R.MBR

**Path:** NXCLOUD/rpgsrc/AP050R.MBR **Generated:** 2026-01-08 15:42:59 **Processing Time:** 15610ms

## Business Logic for Profile Manager API Request

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 how the program constructs requests, handles responses, and manages file operations.

The core logic for sending requests to the Profile Manager API is contained within the main program logic of AP050R. The program prepares a request file, sends it to the API, and processes the response.

## API Request and Response Handling Rules

**AP050R:** Profile Manager API Interaction

### 1. Request Construction

- **Logic:** The program constructs a request file containing user credentials and necessary headers to authenticate with the Profile Manager API.
- **File:** IFS\_Output1 (Request file for API)
- **Field:** w\_ident
- **Condition:** The request file is created if the directory change to the specified path is successful.

### 2. Header File Creation

- **Logic:** A headers file is created to include the API key needed for authentication.
- **File:** IFS\_Output2 (Headers file for API request)
- **Field:** w\_tokeut
- **Condition:** The headers file is created only if the directory change to the specified path is successful.

## API Interaction and Error Handling Rules

### 1. API Call Execution

- **Logic:** The program calls an external web service to send the request and receive a response.
- **Files:**
  - IFS\_Output1 (Request file)
  - IFS\_Input (Response file)
- **Fields:**
  - w\_url2 (API endpoint URL)
  - w\_reqt (HTTP request type)
- **Condition:** The API call is made with the constructed request and headers; the program checks for the response file's existence afterward.

### 2. Error Handling

- **Logic:** If the API call fails, the program sets an error status and proceeds to cleanup.
- **File:** IFS\_Input (Response file)

- **Condition:** If the response file cannot be opened, the program sets API\_status to '9' indicating an error.

-

## File Cleanup Rules

### 1. Temporary File Cleanup

- **Logic:** The program attempts to delete temporary request and response files after processing.

#### • **Files:**

- IFS\_Output1 (Request file)
- IFS\_Output2 (Headers file)
- IFS\_Input (Response file)

- **Condition:** Cleanup is performed if the program logic allows for it, ensuring no residual files remain.

### 2. Final Status Assignment

- **Logic:** The program assigns the final status of the API call to the output parameter.

#### • **File:** None

- **Condition:** The final status is set based on the success or failure of the API interaction.

-

## Special Conditions (Program-Specific)

### 1. Initialization Routine (AP050R)

- **Logic:** The program initializes parameters and retrieves necessary configuration values from the database.

#### • **File:** afpspf and aposextnst (Configuration files)

#### • **Fields:**

- w\_url (API URL)
- w\_token (API key)

- **Condition:** The program fetches values based on the provided firm and system identifiers.

### 2. Error Handling Routine (AP050R)

- **Logic:** This routine is invoked when an error occurs during the API call, setting an error status.

#### • **File:** None

- **Condition:** The program jumps to this routine if the API call fails, ensuring proper error management.

-

## 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 after constructing the request and headers.

- **Logic:** This program handles the actual HTTP request to the Profile Manager API.

- **Impact:** This call represents a **critical step** in the process, as it directly interacts with the external API.

### 2. AP051R (Response Processing)

- **Trigger:** Called if the API call is successful and the response file is available.

- Logic:** This program processes the response data received from the API.
- Impact:** This step is essential for extracting and utilizing the profile information returned by the API.

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

- Trigger:** Called during initialization to generate a unique number for the request.
- Logic:** This program generates a sequential number used in the request and response file naming.
- Impact:** This ensures that each request is uniquely identified, which is crucial for tracking and processing responses.