**Process Steps of the AR415P RPGLE Program**

The AR415P program is a workstation-based interactive program that displays a screen to capture a company code, validates it, and sets up conditions for the subsequent report generation. Here’s a step-by-step breakdown of the process:

1. **Program Initialization**:
   * The program uses the DFTACTGRP(\*NO) directive, indicating it runs in a named activation group, and DFTNAME(AR415P) sets the default program name.
   * The FIXNBR(\*ZONED:\*INPUTPACKED) option ensures zoned decimal and packed decimal fields are handled correctly during input.
   * Files are defined:
     + AR415PD (workstation file, likely a display file for screen interaction, using the Profound UI handler).
     + BICONT (input file, 256 bytes, keyed at position 2).
     + GSCONT (input file, 512 bytes, keyed at position 2).
   * Data structures and variables are defined:
     + msg array for error messages.
     + @infds for workstation file information (e.g., function key status).
     + uds data structure with co (company code, 2-digit numeric) and kycanc (cancel key status, 6 characters).
     + Input specifications for BICONT (field bcdel at position 1) and GSCONT (fields GXDEL at position 1, GXCONO at positions 77–78).
2. **Main Processing Loop (dow @sfnex <> 'EJ')**
   * The program enters a DO loop that continues until @sfnex (screen exit flag) equals 'EJ' (end job).
   * The loop processes the screen based on the @sfid (screen format ID) value:
     + If @sfid is blank, execute subroutine $sblk (initial blank screen).
     + If @sfid is 'S1', execute subroutine $s1 (process screen input).
   * The $xcpt subroutine is called to handle screen display.
   * The program reads the AR415S1 format (from AR415PD display file) with indicators 50 (error) or LR (last record), depending on whether it’s the first screen (@ccnt = 1) or subsequent iterations.
   * The loop ends when @sfnex = 'EJ' (set in $s1ent or $s1ck), triggering the endit tag and setting \*INLR to \*ON to end the program.
3. **Subroutine $sblk: Initial Blank Screen**
   * Executed when @sfid is blank (first time only).
   * Sets indicator \*IN99 to \*ON to display the blank screen.
   * Retrieves the company code (GXCONO) from the GSCONT file using a CHAIN operation (keyed on '00').
     + If found (\*IN95 = \*OFF) and GXCONO is non-zero, sets co = GXCONO.
   * Sets @sfnex and @sfid to 'S1' to display the AR415S1 screen format next.
4. **Subroutine $s1: Process Screen Input**
   * Handles input from the AR415S1 screen format.
   * Checks the function key pressed (via @vkey in @infds):
     + If @vkey = 0 (Enter key), calls $s1ent to validate the entered company code.
     + If @vkey = 2 (Command key, likely F3 or Cancel), calls $s1ck to handle cancellation.
5. **Subroutine $s1ent: Validate Company Code**
   * Validates the co field entered by the user:
     + If co = 0, sets error message msg(1) (“INVALID COMPANY”) and indicator \*IN90 to \*ON, then jumps to ends1e to redisplay the screen.
     + Performs a CHAIN to BICONT using co as the key:
       - If not found (\*IN96 = \*ON), sets error message msg(1) (“INVALID COMPANY”) and \*IN90 to \*ON.
       - If found but bcdel = 'D' (deleted company), sets error message msg(2) (“COMPANY HAS BEEN DELETED”) and \*IN90 to \*ON.
     + If validation passes (valid, non-deleted company), sets @sfnex = 'EJ' to exit the program.
   * The ends1e label ensures the screen is redisplayed with error messages if validation fails.
6. **Subroutine $s1ck: Handle Command Key**
   * Processes command key actions (e.g., F3 for Cancel).
   * If \*INKG (F3 key) is \*ON, sets @sfnex = 'EJ' to exit the program and kycanc = 'CANCEL' to indicate cancellation.
   * Ends with the ends1c label.
7. **Subroutine $xcpt: Display Screen**
   * Increments @ccnt (screen counter).
   * If @sfnex = @sfid, sets \*IN98 to \*ON to indicate the same screen is being redisplayed.
   * If @sfnex = 'S1', writes the AR415S1 format to the display file to show the screen.
   * Calls clrind to clear indicators and error messages.
8. **Subroutine clrind: Clear Indicators**
   * Resets indicators \*IN90, \*IN81, \*IN82, \*IN12, \*IN13, \*IN14, and \*IN15 to \*OFF.
   * Clears the msg1 field to ensure no residual error messages are displayed.
9. **Program Termination**:
   * When @sfnex = 'EJ', the program jumps to the endit tag, sets \*INLR = \*ON, and terminates.

**Business Rules**

The program enforces the following business rules:

1. **Company Code Validation**:
   * The user must enter a valid company code (co) that exists in the BICONT file.
   * The company code must not be zero (invalid company).
   * The company must not be marked as deleted (bcdel <> 'D' in BICONT).
2. **Default Company Code**:
   * If available, the program retrieves a default company code (GXCONO) from the GSCONT file (key '00') for the initial screen.
3. **Error Handling**:
   * Displays “INVALID COMPANY” if the entered co is zero or not found in BICONT.
   * Displays “COMPANY HAS BEEN DELETED” if the company exists but is marked deleted.
   * Errors cause the screen to be redisplayed with the appropriate message.
4. **Cancellation**:
   * The user can cancel the program using a command key (F3, \*INKG), setting kycanc = 'CANCEL' and exiting the program.
5. **Screen Flow**:
   * Starts with a blank screen, then displays the AR415S1 format for company code input.
   * Exits only when a valid company code is entered or the user cancels.

**Tables (Files) Used**

The program uses the following files:

1. **AR415PD** (cf, workstation file):
   * A display file for screen interaction, using the AR415S1 format to capture the company code and display error messages.
   * Managed by the Profound UI handler for modernized UI.
2. **BICONT** (if, input file):
   * A 256-byte file, keyed at position 2, containing company data.
   * Field: bcdel (position 1, 1 byte, indicates if the company is deleted with 'D').
3. **GSCONT** (if, input file):
   * A 512-byte file, keyed at position 2, containing control data.
   * Fields: GXDEL (position 1, 1 byte), GXCONO (positions 77–78, 2-digit numeric company code).

**External Programs Called**

The AR415P program does not explicitly call any external programs. It is an interactive program that processes user input and validates data using file I/O operations. The Profound UI handler (PROFOUNDUI(HANDLER)) is used for the workstation file, but this is a runtime component, not a separate program.

**Summary**

**Process Steps**:

* Initializes the program and displays a blank screen.
* Retrieves a default company code from GSCONT.
* Displays the AR415S1 screen for user input.
* Validates the entered company code against BICONT, checking for existence and deletion status.
* Handles errors by redisplaying the screen with messages or exits on valid input/cancellation.
* Terminates when a valid company code is entered or the user cancels.

**Business Rules**:

* Ensures a non-zero, non-deleted company code is entered.
* Provides default company code from GSCONT.
* Supports cancellation via F3.
* Displays error messages for invalid or deleted companies.

**Files Used**: AR415PD (display), BICONT (company data), GSCONT (control data). **External Programs**: None.