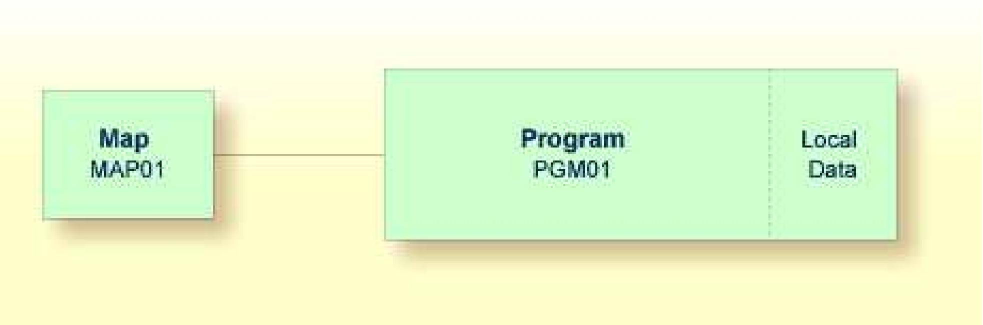
Loops and Labels

You will now enhance your program by adding loops and labels.

When you have completed the exercises below, your sample application will still consist of the same modules as in the previous chapter:



This chapter contains the following exercises:

- Allowing Repeated Usage
- Displaying a Message Indicating that Information was not Found

Allowing Repeated Usage

As it is now, the program terminates after it has displayed the map and has shown the list. So that the user can display a new employees list immediately, without restarting the program, you will now put the corresponding program code into a REPEAT loop.

To define a repeat loop

1. Insert the following below END-DEFINE:

   ```
   RP1. REPEAT
   ```

   REPEAT defines the start of the repeat loop. RP1. is a label which is used when leaving the repeat loop (this is defined below).

2. Define the end of the repeat loop by inserting the following before the END statement:

   ```
   END-REPEAT
   ```

3. Insert the following below INPUT USING MAP 'MAP01':

   ```
   IF #NAME-START = '.' THEN
     ESCAPE BOTTOM (RP1.)
   END-IF
   ```

   The IF statement, which must be ended with END-IF, checks the content of the #NAME-START field. When a dot (.) is entered in this field, the ESCAPE BOTTOM statement is used to leave the loop. Processing will continue with the first statement following the loop (which is END in this case).

By assigning a label to the loop (here `RP1.`), you can refer to this specific loop in the `ESCAPE BOTTOM` statement. Since loops may be nested, you should specify which loop you want to leave. Otherwise, the program will only leave the innermost active loop.

Your program should now look as follows:

```
DEFINE DATA
LOCAL
  1 #NAME-START        (A20)
  1 #NAME-END          (A20)
  1 EMPLOYEES-VIEW VIEW OF EMPLOYEES
    2 FULL-NAME
      3 NAME (A20)
    2 DEPT (A6)
    2 LEAVE-DATA
      3 LEAVE-DUE (N2)
END-DEFINE
*
RP1. REPEAT
*
  INPUT USING MAP ’MAP01’
*
  IF #NAME-START = ’.’ THEN
    ESCAPE BOTTOM (RP1.)
  END-IF
*
  IF #NAME-END = ’ ’ THEN
    MOVE #NAME-START TO #NAME-END
  END-IF
*
  READ EMPLOYEES-VIEW BY NAME
    STARTING FROM #NAME-START
    ENDING AT #NAME-END
*
    DISPLAY NAME 3X DEPT 3X LEAVE-DUE
*
  END-READ
*
END-REPEAT
*
END
```

Note:
For better readability, the content of the `REPEAT` loop has been indented.

4. Run the program.

5. In the resulting map, enter "JONES" in the field which prompts for a starting name and press ENTER.

   In the resulting list, the employees with the name "Jones" are shown. Press ENTER. Due to the `REPEAT` loop, the map is shown again. Now you can also see that "JONES" has been entered as the ending name.

6. To leave the map, enter a dot (.) in the field which prompts for a starting name and press ENTER. Do not forget to delete the remaining characters of the name which is still shown in this field.

7. Stow the program.

Displaying a Message Indicating that Information was not Found

You will now define the message that is to be displayed when the user enters a starting name which cannot be found in the database.

To define the message that is to be displayed when the specified employee cannot be found

1. Add the label `RD1.` to the line containing the `READ` statement so that it looks as follows:

```
RD1. READ EMPLOYEES-VIEW BY NAME
```

2. Insert the following below `END-READ`:

```
IF *COUNTER (RD1.) = 0 THEN
   REINPUT ’No employees meet your criteria.’
END-IF
```

To check the number of records found in the `READ` loop, the system variable `*COUNTER` is used. If its contents equals 0 (that is: an employee with the specified name has not been found), the message defined with the `REINPUT` statement is displayed at the bottom of your map.

To identify the `READ` loop, you assign a label to it (here `RD1.`). Since a complex database access program can contain many loops, you have to specify the loop to which you refer.

Your program should now look as follows:

```
DEFINE DATA
LOCAL
  1 #NAME-START        (A20)
  1 #NAME-END          (A20)
  1 EMPLOYEES-VIEW VIEW OF EMPLOYEES
    2 FULL-NAME
      3 NAME (A20)
    2 DEPT (A6)
    2 LEAVE-DATA
      3 LEAVE-DUE (N2)
END-DEFINE
*
RP1. REPEAT
*
  INPUT USING MAP ’MAP01’
*
  IF #NAME-START = ’.’ THEN
    ESCAPE BOTTOM (RP1.)
  END-IF
*
  IF #NAME-END = ’ ’ THEN
    MOVE #NAME-START TO #NAME-END
  END-IF
*
  RD1. READ EMPLOYEES-VIEW BY NAME
    STARTING FROM #NAME-START
    ENDING AT #NAME-END
*
```

```
    DISPLAY NAME 3X DEPT 3X LEAVE-DUE
*
  END-READ
*
  IF *COUNTER (RD1.) = 0 THEN
    REINPUT ’No employees meet your criteria.’
  END-IF
*
END-REPEAT
*
END
```

3. Run the program.

4. In the resulting map, enter a starting name which is not defined in the demo database (for example, "XYZ") and press ENTER.

   Your message should now appear in the map.

5. To leave the map, enter a dot (.) in the field which prompts for a starting name and press ENTER. Do not forget to delete the remaining characters of the name which is still shown in this field.

6. Stow the program.

You can now proceed with the next exercises: *Inline Subroutines*.