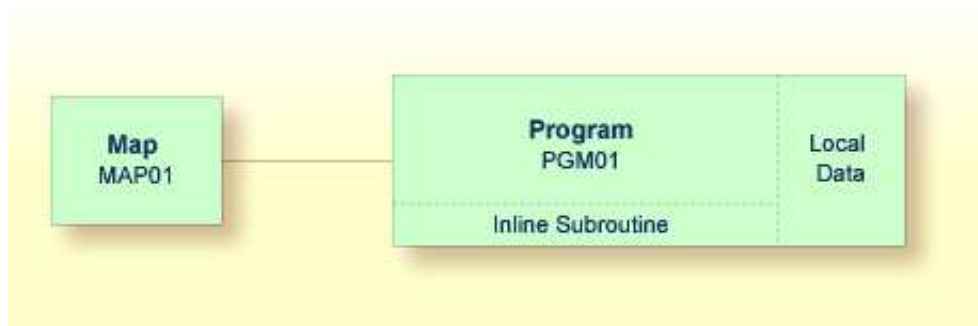


# Inline Subroutines

Natural distinguishes two types of subroutines: inline subroutines which are defined directly in the program and external subroutines which are stored as separate objects outside the program (this is explained later in this tutorial).

You will now add an inline subroutine to your program which moves an asterisk (\*) to the new user-defined variable named #MARK. This subroutine will be invoked when an employee has 20 days of leave or more.

When you have completed the exercises below, your sample application will be structured as follows:



This chapter contains the following exercises:

- Defining the Inline Subroutine
- Performing the Inline Subroutine

---

## Defining the Inline Subroutine

You will now add the subroutine to your program.

### ▶ To define the subroutine

1. Insert the following below the user-defined variable #NAME-END:

```
1 #MARK                (A1)
```

This variable will be used by the subroutine. Therefore, it has to be defined first.

2. To define the subroutine, insert the following before the END statement:

```
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
```

When performed, this subroutine moves an asterisk (\*) to #MARK.

**Note:**

Instead of using the statement `MOVE '*' TO #MARK` it is also possible to use the following variant of the `ASSIGN` or `COMPUTE` statement: `#MARK := '*'`.

3. Modify the `DISPLAY` statement as follows:

```
DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20' #MARK
```

This displays a new column in your output. Its heading is ">=20". The column will contain an asterisk (\*) if the corresponding employee has 20 days of leave or more.

## Performing the Inline Subroutine

Now that you have defined the inline subroutine, you can specify the corresponding code for performing it.

### To perform the inline subroutine

1. Insert the following before the `DISPLAY` statement:

```
IF LEAVE-DUE >= 20 THEN
  PERFORM MARK-SPECIAL-EMPLOYEES
ELSE
  RESET #MARK
END-IF
```

When an employee is found who has 20 days of leave or more, the new subroutine named `MARK-SPECIAL-EMPLOYEES` is performed. When an employee has less than 20 days of leave, the content of `#MARK` is reset to blank.

Your program should now look as follows:

```
DEFINE DATA
LOCAL
  1 #NAME-START          (A20)
  1 #NAME-END            (A20)
  1 #MARK                (A1)
  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
*
IF LEAVE-DUE >= 20 THEN
  PERFORM MARK-SPECIAL-EMPLOYEES
ELSE
  RESET #MARK
END-IF
*
DISPLAY NAME 3X DEPT 3X LEAVE-DUE 3X '>=20' #MARK
*
END-READ
*
IF *COUNTER (RD1.) = 0 THEN
  REINPUT 'No employees meet your criteria.'
END-IF
*
END-REPEAT
*
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
*
END

```

2. Run the program.
3. In the resulting map, enter "JONES" and press ENTER.  
The list of employees should now contain the additional column.
4. To return to the program editor, enter EDIT at the MORE prompt.
5. Stow the program.
6. Enter a dot (.) in the command line to return to the **Development Functions** menu.

You can now proceed with the next exercises: *Processing Rules and Helproutines*.