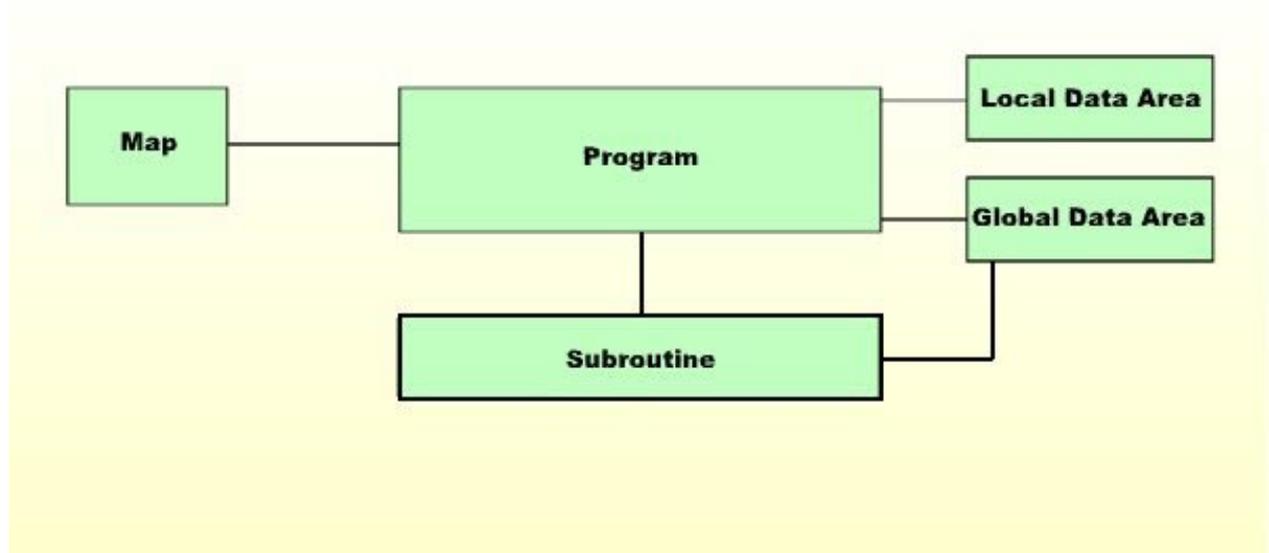


# Session 4 - Creating an External Subroutine

In Natural, a subroutine can be defined either within a program, or as an external subroutine outside the program.

Until now, the subroutine MARK-SPECIAL-EMPLOYEES has been defined within the program using a DEFINE SUBROUTINE statement. In this session, the subroutine will be defined as a separate object external to the program.

Because both internal and external subroutines are invoked with a PERFORM statement, only minimal changes to the program are required.



Listed below are Steps 1 to 5 of Session 4.

---

## Step 1

If Program PGM01 is not already in the program editor, on the Development Functions menu, enter the code **E** and name PGM01.

Make a copy of it by saving it under a different name: at the command prompt of the editor, enter the command SAVE SUBR01.

Enter the command READ SUBR01 to read the new copy into the work area of the editor.

Enter the command SET TYPE S to change the object type from program to subroutine.

## Step 2

Delete all lines of the subroutine except the comment lines, the DEFINE DATA and DEFINE SUBROUTINE blocks, and the END statement, so that the program looks like the illustration below.

You can delete lines quickly by marking the first line of a block of lines with the line command **.X** and the last line of a block with **.Y** and then entering the command **DX-Y** at the command prompt to delete the specified block. Add a comment to identify this subroutine.

**Subroutine SUBR01:**

```

* Example Subroutine: SUBR01
* *****
DEFINE DATA
  GLOBAL USING GDA01
  LOCAL  USING LDA01
END-DEFINE
*
DEFINE SUBROUTINE MARK-SPECIAL-EMPLOYEES
  MOVE '*' TO #MARK
END-SUBROUTINE
END

```

CHECK your changes and correct any errors. Then STOW the subroutine.

## Step 3

Now that the subroutine is located in SUBR01, the internal subroutine must be removed from Program PGM01.

At the command prompt of the editor, enter READ PGM01.

## Step 4

At the command prompt, enter the command BOTTOM to move to the subroutine definition at the end of the program. Delete the following lines, containing the internal subroutine definition, from the program:

```

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

```

The program should now look like the example below:

**Program PGM01**

```

* Example Program 'PGM01' for Natural Tutorial
* PROGRAM NOW USES A LOCAL DATA AREA
* A GLOBAL DATA AREA AND TITLE HAVE BEEN ADDED AND
* THE DISPLAY STATEMENT HAS BEEN CHANGED
* THE SUBROUTINE IS NOW EXTERNAL
* -----
DEFINE DATA
  GLOBAL USING GDA01
  LOCAL  USING LDA01
END-DEFINE
*
REPEAT
*
  INPUT USING MAP 'MAP01'
  IF #NAME-START = '.'
    ESCAPE BOTTOM
  END-IF
  MOVE #NAME-START TO #NAME-END
*
  RD1. READ EMPLOYEES-VIEW BY NAME
        STARTING FROM #NAME-START
        THRU #NAME-END
*
  IF LEAVE-DUE >= 20

```

```

        PERFORM MARK-SPECIAL-EMPLOYEES
    ELSE
        RESET #MARK
    END-IF
*
WRITE TITLE / '*** PERSONS WITH 20 OR MORE DAYS LEAVE DUE ***'
           / '*** ARE MARKED WITH AN ASTERISK ***' //
*
DISPLAY 23X '//N A M E' NAME
          3X '//DEPT'   DEPT
          3X '//LV/DUE' LEAVE-DUE
          3X '//*'     #MARK
*
END-READ
*
IF *COUNTER (RD1.) = 0
    REINPUT 'PLEASE TRY ANOTHER NAME'
END-IF
*
END-REPEAT
END

```

## Step 5

CHECK the program and correct any errors. Then RUN the program to make sure that the results are the same with an external subroutine as previously with an internal subroutine.

STOW the program for the next session.

At the command prompt, then enter a period (.) to return to the Development Functions menu.

End of Session 4.