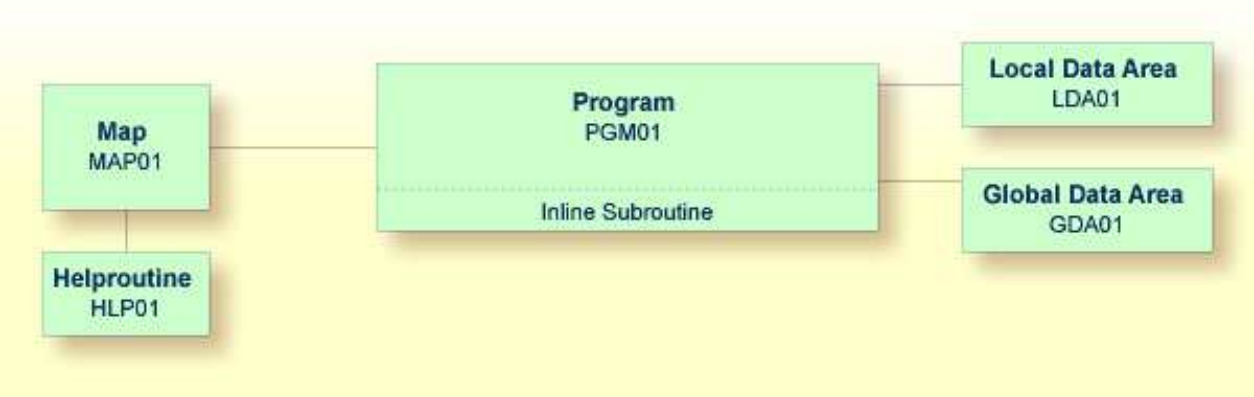
Global Data Areas

Data defined in a global data area (GDA) can be shared by multiple programs, external subroutines and helproutines.

Any modification of a data element value in a global data area affects all Natural objects that reference this global data area. Therefore, if you change the source of a global data area, you have to stow all previously created Natural objects that reference this global data area once more. The sequence in which objects are stowed is important. You must first stow the global data area and then the program. If you stow the program first and then the global data area, the program cannot be stowed because new elements in the global data area cannot be found.

You will now create a global data area which will be shared by your program and an external subroutine that you will create later. As the basis for your global data area, you will use some of the information from the local data area you have just created.

When you have completed the exercises below, your sample application will consist of the following modules:



This chapter contains the following exercises:

- Creating a Global Data Area from an Existing Local Data Area
- Adapting the Local Data Area
- Referencing the Global Data Area from Your Program

Creating a Global Data Area from an Existing Local Data Area

You can create a new data area from an existing data area by editing it and saving it under a different name and with a different type. The original data area remains unchanged, and the new data area can be edited. Since the fields `#NAME-START` and `#NAME-END` are not required in the global data area, you will remove them.

To create the global data area

1. Return to your local data area by entering the following in the command line of the program editor.

```
E LDA01
```

2. To save the data area under a new name, enter the following in the command line of the data area editor.

```
SA GDA01
```

   The current data area is saved with the new name `GDA01`. The local data area named `LDA01` is still shown in the data area editor.

3. Load `GDA01` into the data area editor by entering the following command:

```
E GDA01
```

4. To change the local data area into a global data area, enter the following command:

```
SET TYPE G
```

   where "G" denotes global data area.

   The object type changes to "Global". This is indicated at the top of the screen.

5. Press ESC to enter edit mode. Use the line command `D` to delete the following fields:

   ```
   #NAME-START
   #NAME-END
   ```

6. The global data area should now look as follows:

```
                      Press <ESC> to enter command mode
 Mem: GDA01    Lib: TUTORIAL Type: GLOBAL     Bytes:   351  Line:    1 of:   8
C T L  Name of Datafield                F     Length Index/Comment             M
  *    *** Top of Data Area ***
     1 #MARK                            A          1
  V  1 EMPLOYEES-VIEW                                EMPLOYEES
     2 PERSONNEL-ID                     A          8
  G  2 FULL-NAME
     3 NAME                             A         20
     2 DEPT                             A          6
  G  2 LEAVE-DATA
     3 LEAVE-DUE                        N          2
  *    *** End of Data Area ***









  F 1 HELP     F 2 CHOICE   F 3 QUIT     F 4 SAVE     F 5 STOW     F 6 CHECK
  F 7 READ     F 8 CLEAR    F 9 MEM TYPE F10 GEN      F11 FLD TYPE F12
```

7. Stow the global data area.

Adapting the Local Data Area

The fields contained in the global data area are no longer required in the local data area. Therefore, you will now remove all fields except #NAME-START and #NAME-END from the local data area.

To remove the fields

1. Return to your local data area by entering the following in the command line of the data area editor:

   ```
   E LDA01
   ```

2. Use the line command D to delete all fields except #NAME-START and #NAME-END.

   When you delete the top-level entry for the view (indicated by a "V" in front of the view name), all fields belonging to this view are automatically deleted.

3. Stow the modified local data area.

   The local data area should now look as follows:

```
Command:
 Mem: LDA01    Lib: TUTORIAL Type: LOCAL      Bytes:    85  Line:      of:   2
C T    Comment
  *    *** Top of Data Area ***
     1 #NAME-START                      A         20
     1 #NAME-END                        A         20
  *    *** End of Data Area ***









  F 1 HELP     F 2 CHOICE   F 3 QUIT     F 4 SAVE     F 5 STOW     F 6 CHECK
  F 7 READ     F 8 CLEAR    F 9 MEM TYPE F10 GEN      F11 FLD TYPE F12
```

Referencing the Global Data Area from Your Program

Once a global data area has been stowed, it can be referenced by a Natural program.

You will now change the `DEFINE DATA` statement in your program so that it also uses the global data area that you have just defined.

To use the global data area in your program

1. Return to the program editor by entering the following in the command line of the data area editor.

   ```
   E PGM01
   ```

2. Insert the following in the line above `LOCAL USING LDA01`:

   ```
   GLOBAL USING GDA01
   ```

   A global data area must always be defined before a local data area. Otherwise, an error occurs.

   Your program should now look as follows:

   ```
   DEFINE DATA
     GLOBAL USING GDA01
     LOCAL USING LDA01
   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
```

3. Run the program.
4. To confirm that the results are the same as before (when the `DEFINE DATA` statement did not reference a global data area), enter "JONES" as the starting name and press ENTER.
5. To return to the program editor, enter `EDIT` at the `MORE` prompt.
6. Stow the program.

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