Processing Rules and Helproutines

Processing rules and helproutines are defined for fields in a map.

When you have completed the exercises below, your sample application will consist of the following modules (a processing rule cannot be defined as a separate module; it is always part of a map):

This chapter contains the following exercises:

- Defining a Processing Rule

- Defining a Helproutine

Defining a Processing Rule

You will now define the message that is to be displayed when the user presses ENTER without specifying a starting name.

▶ **To define a processing rule**

1. Return to the map editor by entering the following in the **Development Functions** menu.

```
           Code .. E    Type .. _
                        Name .. MAP01_____
```

2. Starting in the first position of the input field for the starting name, enter the following:

```
 .P
```

Example:

```
              Start .PXXXXXXXXXXXXXXXXXXX

              End :XXXXXXXXXXXXXXXXXXX
```

3. Press ENTER

    The following screen is displayed for the selected field:

```
Variables used in current map                                              Mod
#NAME-START(A20)                                                             U
#NAME-END(A20)                                                               U




Rule _____ Field #NAME-START
>                                      > + Rank 0    S     L 1     Struct Mode
ALL    ....+....10...+....+....+....30...+....+....+....50...+....+....+....70.
  0010
  0020
  0030
  0040
  0050
  0060
  0070
  0080
  0090
  0100
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Mset  Exit  Test         --    -     +     Full  Sc=         Let
```

4. Enter the following processing rule:

```
IF & = ' ' THEN
  REINPUT 'Please enter a starting name.'
  MARK *&
END-IF
```

The ampersand (&) in the processing rule will dynamically be replaced with the name of the field. In this case, it will be replaced with #NAME-START. If #NAME-START is blank, the message defined with the REINPUT statement is displayed.

MARK is an option of the REINPUT statement. Its syntax is MARK *fieldname. MARK specifies the field in which the cursor is to be placed when the REINPUT statement is executed. In this case, the cursor will be placed in the #NAME-START field.

5. In the **Rank** field, enter "1".

```
Rule _____ Field #NAME-START
>                                      > + Rank 1    S 2    L 1    Struct Mode
ALL    ....+....10...+....+....+....30...+....+....+....50...+....+....+....70.
  0010 IF & = ' ' THEN
  0020   REINPUT 'Please enter a starting name.'
  0030   MARK *&
  0040 END-IF
  0050
```

The rank defines the sequence in which the rules for the different fields are to be processed. All rules with rank 1 are processed first, followed by those with rank 2, etc.

6. Press ENTER to save your input. Then press PF3 to return to the map.

   Note:
   If you want to redisplay your processing rule, you have to use the command `.P1` (to display the rule with the rank 1) or `.P*` (to display a list of all rules defined for this field).

7. Test the map.

8. In the resulting output, enter any starting name and press ENTER.

   The output screen is closed.

9. Test the map once more. Do not enter a name and press ENTER.

   The message defined with the processing rule should now appear in the map.

10. To leave the output screen, enter a dot (.) in the field which prompts for a starting name and press ENTER.

11. Stow the map (press PF3 to return to the **Edit Map** menu and enter "W" in the **Code** field).

Defining a Helproutine

A helproutine is displayed when the user presses the help key when the cursor is on the input field for the starting name.

You will first define the helproutine and then associate it with a specific field.

▶ **To create a helproutine**

1. Return to the **Development Functions** menu by pressing PF3 in the **Edit Map** menu.

2. At the bottom of the **Development Functions** menu, enter the following information and press ENTER:

```
                Code .. C    Type .. H
                             Name .. HLP01_____
```

"C" stands for the function **Create Object**, "H" stands for the object type helproutine, and "HLP01" is the name of your new helproutine.

An empty editor appears.

3. Enter the following:

```
WRITE 'Type the name of an employee'
END
```

4. Stow the helproutine.

▶ **To associate the helproutine with a field on the map**

1. Return to the map editor by entering the following in the command line of the screen in which you have just entered the helproutine.

```
  E MAP01
```

2. Starting in the first position of the data field for the starting name, enter the following and press ENTER:

```
  .E
```

Or:
Position the cursor anywhere in the data field and press PF5.

The extended field editing section is displayed for the field.

3. In the **HE** field enter "'HLP01'" (including the single quotation marks).

This is the name under which you have saved your helproutine.

```
Fld #NAME-START                                                        Fmt A20
--------------------------------------------------------------------------------
AD= MIT'_'_____       ZP=          SG=          HE= 'HLP01'_____   Rls 2
AL= _____            CD= __       CV= _____   Mod User
PM= __  DF=                        DY= _____
EM= _____     SB= _____
```

4. Press PF3 to leave the extended field editing section.

5. Test the map.

6. In the resulting output, enter a question mark (?) in the input field for the starting name and press ENTER.

The help text you have defined is shown.

7. Press ENTER to return to the map.

8. To leave the map, enter a dot (.) in the field which prompts for a starting name and press ENTER.

9. Stow the map (press PF3 to return to the **Edit Map** menu and enter "W" in the **Code** field).

10. Press PF3 to return to the **Development Functions** menu.

You can now proceed with the next exercises: *Local Data Areas*.