

# MODULE POOL SUB SCREEN

## (ADDITIN, SUBTRACTION, MULTIPLICATION).

Here I'm adding only the code, screens and what are all the fields I'm using....

First write a code inside the include function **INCLUDE ZTESTTOP** if we double click on the highlighted one then we can see the below code.

Here we are declaring the fields...**code**

**PROGRAM ZSCREEN\_TEST.**

**\*INCLUDE ZTESTTOP**

**DATA : num1(10) TYPE c,**

**num2(10) TYPE c,**

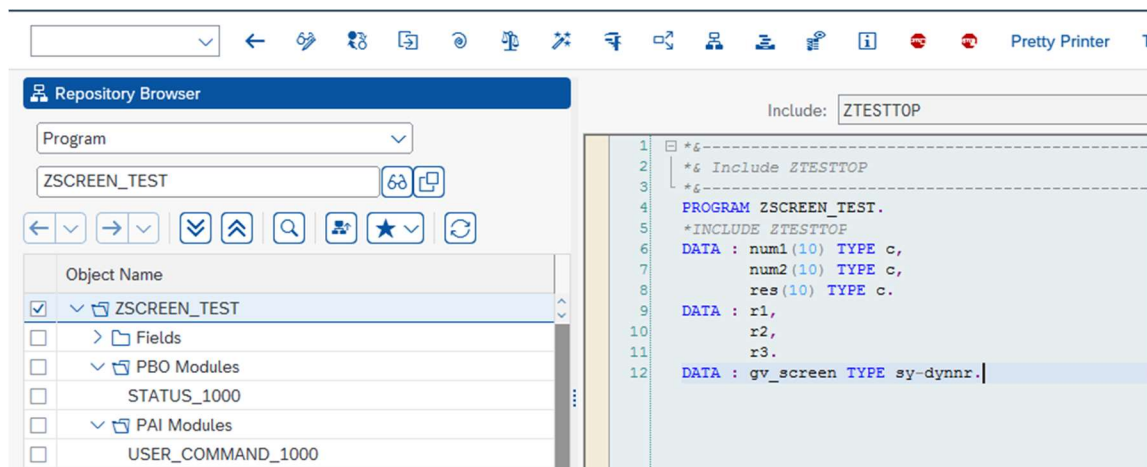
**res(10) TYPE c.**

**DATA : r1,**

**r2,**

**r3.**

**DATA : gv\_screen TYPE sy-dynnr.**



After declaring the key words, variables, fields then we can write a logic under the screen so for that..

Create a screen as main screen with short description then the PBO and PAI will open then go to layout.

After that use the input /output field , push button, box , radio buttons, and also here mainly using the **sub screen** in scree using all these things then click on flow logic..

The image displays the SAP ABAP development environment, specifically the Screen Painter tool, used for creating and editing screens.

**Top Panel (Repository Browser):** Shows the object hierarchy. The 'ZMODULE\_POOL' object is selected, and the 'Screen' option is highlighted in the context menu. The 'Include: ZMODULE\_POOL\_TOP' is active.

**Attributes Panel:** Displays the screen number '100' and the short description 'SCREEN100'. The 'Dynpro Type' is set to 'Normal'.

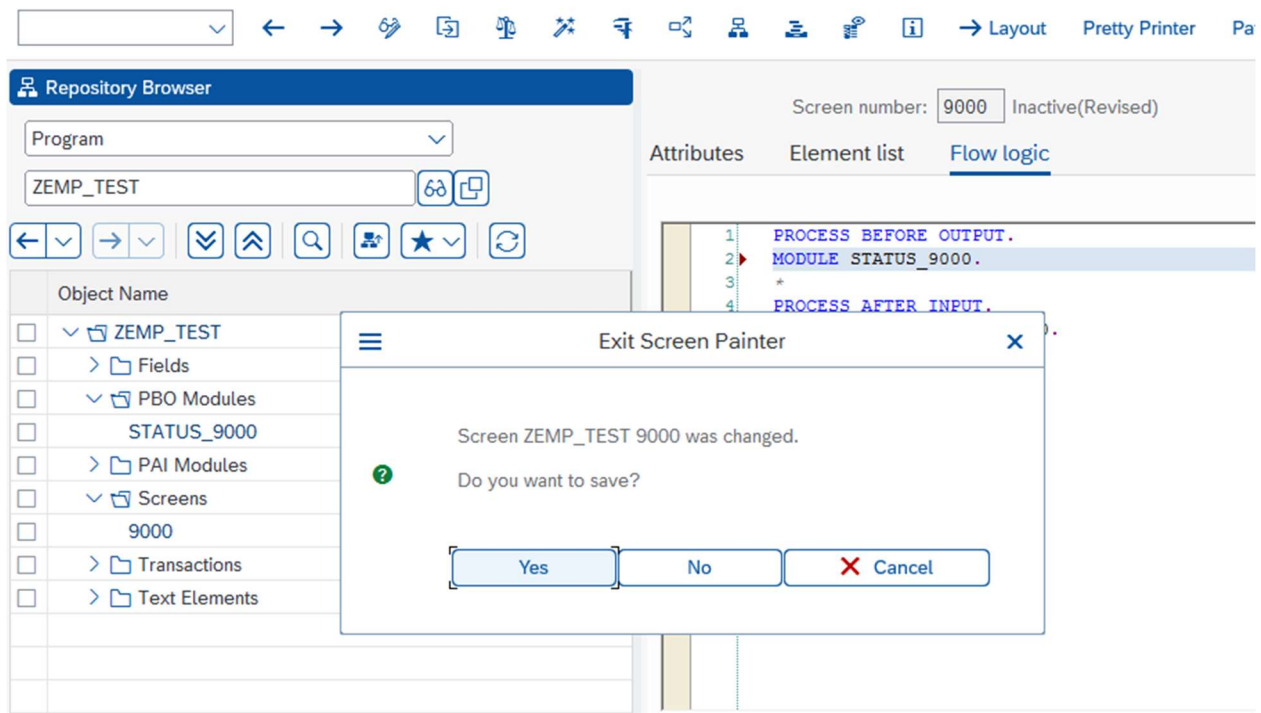
**Flow Logic Panel:** Shows the logic for the screen. The logic includes:

```
1 PROCESS BEFORE OUTPUT.  
2 MODULE STATUS_1000.  
3 CALL SUBSCREEN sub INCLUDING sy-REPID gv_screen.  
4  
5  
6 PROCESS AFTER INPUT.  
7 MODULE USER_COMMAND_1000.  
8 CALL SUBSCREEN sub.
```

**Layout Panel:** Shows the layout of the screen. The 'Flow Logic' tab is selected, and the logic is displayed in the 'Text' area.

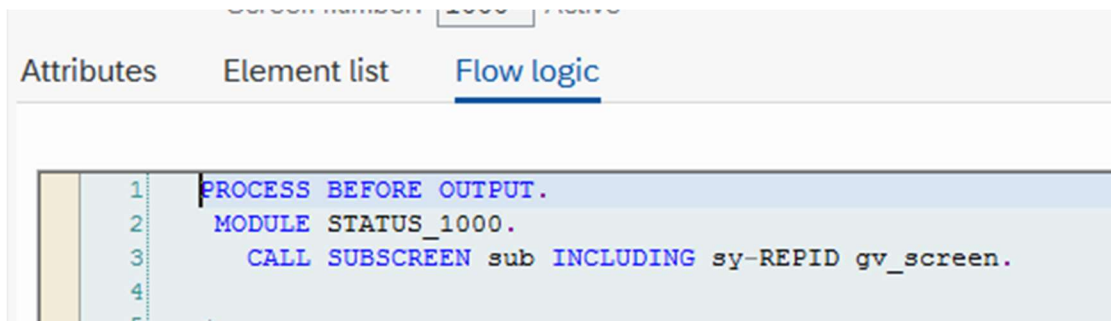
**Bottom Panel (Screen Painter):** Displays the screen 'ZSCREEN\_TEST 1000'. The 'Flow Logic' tab is selected, and the logic is displayed in the 'Text' area. The screen is titled 'Screen Painter: Display Screen ZSCREEN\_TEST 1000'.

Here change the PBO and PAI logic from comment to uncomment. Then double click on the uncommented one. Under that write a logic of the program.



Under the PBO we have to call the sub screen using the **sy-repid**

Here the sub is the functional code and dynamic screen using for main screen using sub screen.



Code(for calling the sub screen) :

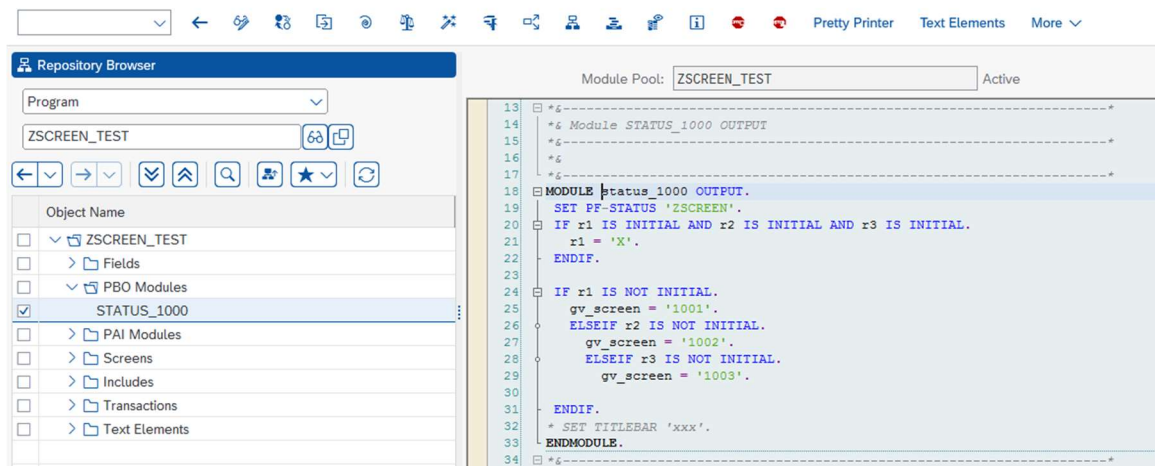
**PROCESS BEFORE OUTPUT.**

**MODULE STATUS\_1000.**

**CALL SUBSCREEN sub INCLUDING sy-REPID gv\_screen.**

## PBO :

Under the PBO written the logic of initializing the radio buttons, and using the dynamics screens we can use the sub screens of the addition, subtraction, multiplication.



## PBO CODE(For Initialization of Sub screen using the radio buttons)

**MODULE status\_1000 OUTPUT.**

**SET PF-STATUS 'ZSCREEN'.**

**IF r1 IS INITIAL AND r2 IS INITIAL AND r3 IS INITIAL.**

**r1 = 'X'.**

**ENDIF.**

**IF r1 IS NOT INITIAL.**

**gv\_screen = '1001'.**

**ELSEIF r2 IS NOT INITIAL.**

**gv\_screen = '1002'.**

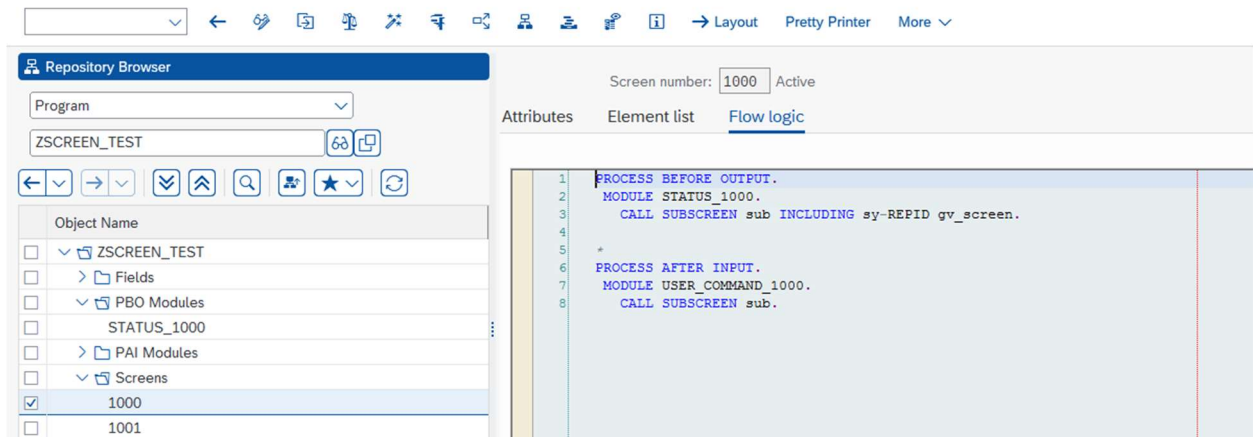
**ELSEIF r3 IS NOT INITIAL.**

**gv\_screen = '1003'.**

**ENDIF.**

**\* SET TITLEBAR 'xxx'.**

**ENDMODULE.**



Under the PAI we have to call the sub screen using the functional code what we are declared in layout .

```
6 PROCESS AFTER INPUT.  
7   MODULE USER_COMMAND_1000.  
8     CALL SUBSCREEN sub.
```

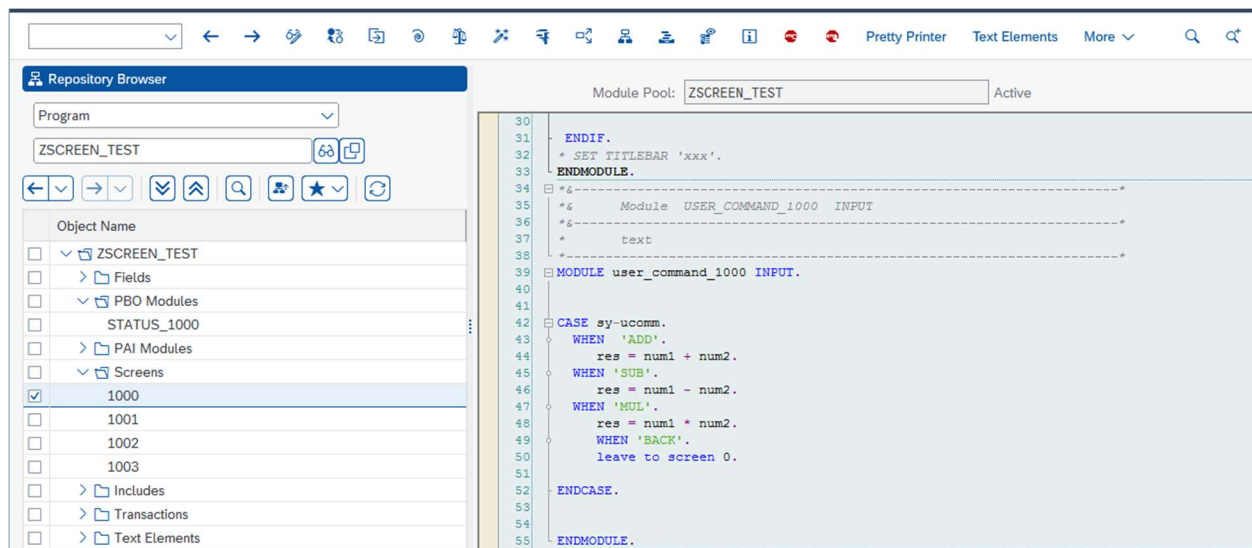
**Code: (for calling the sub screen using functional code)**

**PROCESS AFTER INPUT.**

**MODULE USER\_COMMAND\_1000.**

**CALL SUBSCREEN sub.**

**PAI:** here we are using the case statement for addition, subtraction, multiplication with **sy-ucomm**. And after this we have to go back for that using the BACK button. Then all the process done leave the screen so for that purpose we use the leave screen 0.



**CODE:( user command )**

**MODULE user\_command\_1000 INPUT.**

**CASE sy-ucomm.**

**WHEN 'ADD'.**

**res = num1 + num2.**

**WHEN 'SUB'.**

**res = num1 - num2.**

**WHEN 'MUL'.**

**res = num1 \* num2.**

**WHEN 'BACK'.**

**leave to screen 0.**

**ENDCASE.**

**ENDMODULE.**

Done with all logic in the program active the main program and

The top screenshot shows the SAP ABAP editor with the Repository Browser on the left and the code editor on the right. The Repository Browser shows the object ZSCREEN\_TEST selected. The code editor shows the following code:

```

1  *%-----*
2  *% Module Pool      ZSCREEN_TEST
3  *%-----*
4  *%
5  *%-----*
6
7  INCLUDE ZTESTTOP                      .    " Global Data
8
9  * INCLUDE ZTEST001                      .    " PBO-Modules
10 * INCLUDE ZTESTI01                      .    " PAI-Modules
11 * INCLUDE ZTESTF01                      .    " FORM-Routines
12
13 *%-----*
14 *% Module STATUS_1000 OUTPUT
15 *%-----*
16 *%
17 *%-----*
18 MODULE status_1000 OUTPUT.
19   SET PF-STATUS 'ZSCREEN'.
20   IF r1 IS INITIAL AND r2 IS INITIAL AND r3 IS INITIAL.
21     r1 = 'X'.
22   ENDIF.
23
24   IF r1 IS NOT INITIAL.
25     gv_screen = '1001'.
26     ELSEIF r2 IS NOT INITIAL.

```

The bottom screenshot shows the continuation of the code in the code editor:

```

27   gv_screen = '1002'.
28   ELSEIF r3 IS NOT INITIAL.
29     gv_screen = '1003'.
30
31   ENDIF.
32   * SET TITLEBAR 'xxx'.
33   ENDMODULE.
34
35 *%-----*
36 *%      Module USER_COMMAND_1000  INPUT
37 *%-----*
38 *%
39 *%-----*
40 MODULE user_command_1000 INPUT.
41 CASE sy-ucomm.
42   WHEN 'ADD'.
43     res = num1 + num2.
44   WHEN 'SUB'.
45     res = num1 - num2.
46   WHEN 'MUL'.
47     res = num1 * num2.
48   WHEN 'BACK'.
49     leave to screen 0.
50   ENDCASE.
51
52 ENDMODULE.

```

Main program code : ( includes all the logic in module pool)

CODE:

```

*&-----*
*& Module Pool      ZSCREEN_TEST
*&-----*
*&
*&-----*

```

```
INCLUDE ZTESTTOP          . " Global Data

* INCLUDE ZTESTO01        . " PBO-Modules
* INCLUDE ZTESTI01        . " PAI-Modules
* INCLUDE ZTESTF01        . " FORM-Routines
```

```
*&-----*
```

```
*& Module STATUS_1000 OUTPUT
```

```
*&-----*
```

```
*&
```

```
*&-----*
```

```
MODULE status_1000 OUTPUT.
```

```
SET PF-STATUS 'ZSCREEN'.
```

```
IF r1 IS INITIAL AND r2 IS INITIAL AND r3 IS INITIAL.
```

```
    r1 = 'X'.
```

```
ENDIF.
```

```
IF r1 IS NOT INITIAL.
```

```
    gv_screen = '1001'.
```

```
ELSEIF r2 IS NOT INITIAL.
```

```
    gv_screen = '1002'.
```

```
ELSEIF r3 IS NOT INITIAL.
```

```
    gv_screen = '1003'.
```

```
ENDIF.
```

```
* SET TITLEBAR 'xxx'.
```



ENDMODULE.

\*&-----\*

\*&   Module USER\_COMMAND\_1000 INPUT

\*&-----\*

\*    text

\*-----\*

MODULE user\_command\_1000 INPUT.

CASE sy-ucomm.

  WHEN 'ADD'.

    res = num1 + num2.

  WHEN 'SUB'.

    res = num1 - num2.

  WHEN 'MUL'.

    res = num1 \* num2.

  WHEN 'BACK'.

    leave to screen 0.

ENDCASE.

ENDMODULE.



MODULE POOL  
MAIN PRG.txt



PBO & PAI.txt



INCLUDE PRG.txt

COMPLETED 😊😊