

Dinamik Screen Objeleri



Dinamik Objeler



Buton ile Kullanım



Inputfield ile Kullanım

```
5  *&-----
6  REPORT Z_BK_EGT_18.
7
8
9  data: gv_name type char20,
10         gv_surname type char30,
11         gv_age type num4.
12
13  START-OF-SELECTION.
14  call SCREEN 0100.
15
16  *&-----
17  *& Module STATUS_0100 OUTPUT
18  *&-----
19  *&
20  *&-----
21  MODULE status_0100 OUTPUT.
22  SET PF-STATUS '0100'.
23  * SET TITLEBAR 'xxx'.
24  ENDMODULE.
25
26  *&-----
27  *& Module USER_COMMAND_0100 INPUT
28  *&-----
29  * text
30  *-----
31  MODULE user_command_0100 INPUT.
32  CASE sy-ucomm.
33  WHEN '&BACK'.
34      SET SCREEN 0.
35  WHEN OTHERS.
36  ENDCASE.
37  ENDMODULE.
```

Temel ayarlamaları yaptıktan sonra 3 değişken tanımlayıp ardından ekrandan obje haline getirmiş olduk.

Şimdi burada temelde oynayacağımız 3 özellik var;

Birincisi aktif özelliği.

Invisible özelliği.

Input özelliği. Eğer bu butonsa tıklanabilir özelliğini kapatır.

The screenshot displays the ABAP Debugger interface. At the top, a code editor shows the following code:

```
27 LOOP AT SCREEN.  
28 BREAK-POINT.  
29 ENDLOOP.  
30 ENDMODULE.  
31
```

Below the code editor, the debugger window is titled "ABAP Debugger(1) (Exclusive)(vhcalnplci_NPL_00)". It shows the current module as "Z_BK_EGT_18" and the current statement as "STATUS_0100". The "Structures" tab is active, showing the "SCREEN" structure. The structure is defined as "flat, not charlike(316)".

Exp	Component	Val....	Val.	Ch...	Technical Type	Hexadecimal Value
	NAME		NAME		C(132)	4E0041004D0045002000
	GROUP1				C(3)	200020002000
	GROUP2				C(3)	200020002000
	GROUP3				C(3)	200020002000
	GROUP4				C(3)	200020002000
	REQUIRED	0			C(1)	3000
	INPUT	0			C(1)	3000
	OUTPUT	0			C(1)	3000
	INTENSIFIED	0			C(1)	3000
	INVISIBLE	0			C(1)	3000
	LENGTH	00			X(1)	00

Below the structure table, the code editor shows the following code:

```
MODULE status_0100 OUTPUT.  
SET PF-STATUS '0100'.  
* SET TITLEBAR 'xxx'.  
  
LOOP AT SCREEN.  
IF screen-name EQ 'GV_NAME'.  
screen-active = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.  
MODIFY SCREEN.  
ENDIF.  
ENDLOOP.  
ENDMODULE.  
*&-----*
```

Şimdi PBO kısmında daha ekran gelmeden yapacağımız bir test kodlaması var. Yukarıda kodlarda görüldüğü gibi eğer bir loop içerisine atamayı yapıpta screen-name ile gv_name'i eşitleyebiliyorsak o halde devamı olarak aktifliğini kapatabiliyoruz. screen-active = 0 sayesinde o alanı pasif yapmış oluyoruz. MODIFY SCREEN ile commit ediyoruz gibi düşün. Ve sonuç olarak çıktı şu şekilde oluyor;

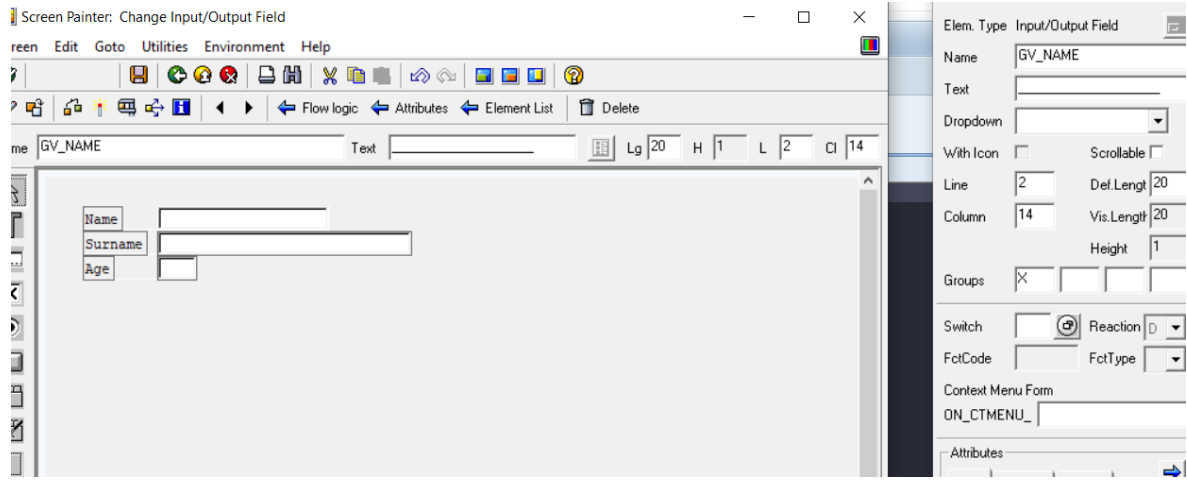
Name	
Surname	<input type="text"/>
Age	<input type="text"/>

NOT: Burada bir etmen var, eğer gv_name'i küçük bırakırsak o kod işlemiyor, GV_NAME olarak göstermek gerekiyor. Yoksa kod yemiyor.

```
3 MODULE status_0100 OUTPUT.  
4   SET PF-STATUS '0100'.  
5   * SET TITLEBAR 'xxx'.  
6  
7   LOOP AT SCREEN.  
8     IF screen-name EQ 'GV_NAME'.  
9       screen-active = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.  
10      MODIFY SCREEN.  
11    ENDIF.  
12  
13    IF screen-name EQ 'GV_SURNAME'.  
14      screen-invisible = 1. "Böylece ekran pasif oluyor, 0 pasif yap demek.  
15      MODIFY SCREEN.  
16    ENDIF.  
17  
18    IF screen-name EQ 'GV_AGE'.  
19      screen-input = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.  
20      MODIFY SCREEN.  
21    ENDIF.  
22  ENDLOOP.  
23 ENDMODULE.
```

Name	
Surname	<input type="text" value="*****"/>
Age	<input type="text" value="0"/>

Ve şöyle de bir şey yapılabilir, burada şu an isim tanımlamaları üzerinden gidiyoruz fakat çok fazla inputbox olabilir ve bunları daha kolay hale getirmek adına gruplandırıyoruz.



Örnek olarak yukarıdaki görselde üç tane input alanı var, iki tanesinin ilk grubuna X sonuncu olan yaşa A atıyoruz ve şu kodları işletiyoruz;

```

MODULE status_0100 OUTPUT.
  SET PF-STATUS '0100'.
  * SET TITLEBAR 'xxx'.

  LOOP AT SCREEN.
    IF screen-group1 EQ 'X'.
      screen-input = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.
      MODIFY SCREEN.
    ENDIF.
    * IF screen-name EQ 'GV_NAME'.
    *   screen-active = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.
    *   MODIFY SCREEN.
    * ENDIF.
    * IF screen-name EQ 'GV_SURNAME'.
    *   screen-invisible = 1. "Böylece ekran pasif oluyor, 0 pasif yap demek.
    *   MODIFY SCREEN.
    * ENDIF.
    * IF screen-name EQ 'GV_AGE'.
    *   screen-input = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.
    *   MODIFY SCREEN.
    * ENDIF.

  ENDLOOP.
ENDMODULE.

```

SAP

Name	<input type="text"/>
Surname	<input type="text"/>
Age	<input type="text"/>

Ve çıktı da bu şekilde olur.

İlk iki inputun ikinci grubuna x sonunda inputun ikinci grubunu boş bırakıp şöyle yaparsak;

Name	<input type="text"/>
Surname	<input type="text"/>
Age	<input type="text"/>

Line	3	Def. Leng	30
Column	14	Vis. Leng	30
		Height	1
Groups	X		
Switch	<input type="checkbox"/>	Reaction	0
FctCode		FctType	

```

MODULE status_0100 OUTPUT.
  SET PF-STATUS '0100'.
  * SET TITLEBAR 'xxx'.

  LOOP AT SCREEN.
    IF screen-group2 EQ 'X'.
      screen-input = 0. "Böylece ekran pasif oluyor, 0 pasif yap demek.
      MODIFY SCREEN.
    ENDIF.
  * IF screen-name EQ 'GV NAME'.

```

SAP

Name	<input type="text"/>
Surname	<input type="text"/>
Age	<input type="text"/>

Nihai olarak diğer grupların kullanımı da bu şekildedir. Her türlü kombinasyonlar yapılarak çeşitlilik sağlanabilir.

Buton ile kullanım

Name	<input type="text"/>
Surname	<input type="text"/>
Age	<input type="text"/>

Görüldüğü gibi enable basınca inputlara giriş açık, disable basınca kapalı olsun istiyoruz. Bundan dolayı da farklı bir durum söz konusu olmuş oluyor.

1.

```

MODULE user_command_0100 INPUT.
CASE sy-ucomm.
  WHEN '&BACK'.
    SET SCREEN 0.
  WHEN '&ENABLE'.
    gv_flag = abap_true. " X
  WHEN '&DISABLE'.
    gv_flag = abap_false. " ' ' ' '
  WHEN OTHERS.
  ENDCASE.
ENDMODULE.

```

2.

```

MODULE status_0100 OUTPUT.
  SET PF-STATUS '0100'.
  * SET TITLEBAR 'xxx'.

  LOOP AT SCREEN.
    IF gv_flag eq abap_true.
      IF screen-group1 eq abap_true.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
    ENDIF.
    IF gv_flag eq abap_false.
      IF screen-group1 eq abap_true.
        screen-input = 0.
        MODIFY SCREEN.
      ENDIF.
    ENDIF.
  ENDIF.
ENDMODULE.

```

3.

Name	<input type="text"/>
Surname	<input type="text"/>
Age	<input type="text"/>

Name	<input type="text"/>
Surname	<input type="text"/>
Age	<input type="text" value="0"/>

INPUTFIELD ile kullanım

Ekranada 5 tane inputfield olacak ve bir tanesiyle diğer dördünü yönetebiliyor olacağız.

Bunun için en başta yeniden bir screen tasarlayıp 0200 koduyla açtık ve eski kodları comment satırına aldık.

Ardından gelenler;

```
MODULE user_command_0200 INPUT.  
  CASE sy-ucomm.  
    WHEN '&BACK'.  
      SET SCREEN 0.  
    ENDCASE.  
  ENDMODULE.
```

```
MODULE status_0200 OUTPUT.  
  SET PF-STATUS '0100'.  
  
  LOOP AT SCREEN.  
    IF screen-group1 EQ abap_true.  
      screen-input = 0.  
      MODIFY SCREEN.  
    ENDIF.  
    IF screen-group2 EQ abap_true.  
      screen-input = 0.  
      MODIFY SCREEN.  
    ENDIF.  
    IF screen-group3 EQ abap_true.  
      screen-input = 0.  
      MODIFY SCREEN.  
    ENDIF.  
    IF screen-group4 EQ abap_true.  
      screen-input = 0.  
      MODIFY SCREEN.  
    ENDIF.  
  ENDLOOP.
```

BU KISIM DIREKT OLARAK BASLANGICTA INPUTLARIN
PASIF DURUMDA OLMASINI SAGLAYAN KODLARDIR.

Inputları daha başlangıçta pasif yaptıktan sonra başka bir loop içerisinde kodlarımıza devam ediyoruz. Bunu yaparken de yine mantıken zaten gruplandırmaya göre gitmiş oluyoruz.

```

LOOP AT SCREEN.
  CASE gv_num.
    WHEN 1.
      IF screen-group1 EQ abap_true.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
    WHEN 2.
      IF screen-group2 EQ abap_true.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
    WHEN 3.
      IF screen-group3 EQ abap_true.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
    WHEN 4.
      IF screen-group4 EQ abap_true.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
  ENDCASE.
ENDLOOP.
ENDMODULE.

```

*c

2