

## ReuseALV\_save-variant-default

```
* I_DEFAULT = 'X'
* I_SAVE = ' '
* IS_VARIANT =
```

Save nedir?

Execute edilen bir projenin üzerinde yapılan değişiklikleri tekrar tekrar aynı şeyleri yapmayalım diye kaydetmeyi sağlayan yapıdır. Örneğin ekranımızda bir tablo mevcut ve yerlerini değiştiriyoruz, kimi sütunları gizliyoruz ve bunu her açtığımızda yapmak yerine bu şekilde kaydedili olarak her açtığımızda son düzenlenmiş haliyle gözükmesini istiyoruz, bu zamanda save kullanılır.

[illegible]

Örneğin yukarıdaki görselde bir sütunu kaldırdık ve bunu bu şekliyle her açtığımızda görmek için oklanmış yere tıklama yaparak kaydediyoruz.





```

*'' REFERENCE(IT_SAVE) DEFAULT SPACE
*'' REFERENCE(IS_VARIANT) LIKE DISVARIANT STRUCTURE DISVARIANT
*'' OPTIONAL
*'' REFERENCE(IT_EVENTS) TYPE SLIS_T_EVENT OPTIONAL
*'' REFERENCE(IT_EVENT_EXIT) TYPE SLIS_T_EVENT_EXIT OPTIONAL
*'' REFERENCE(IS_PRINT) TYPE SLIS_PRINT_ALV OPTIONAL
*'' REFERENCE(IS_REPREP_ID) TYPE SLIS_REPREP_ID OPTIONAL

```

data: gs\_variant type DISVARIANT.

```

data: gs_variant type DISVARIANT.

```

```

I_SAVE = 'X'
IS_VARIANT = gs_variant
it_events = gt_events
* IT_EVENT_EXIT =

```

```

gs_variant-variant = '/BURAK'. "Burada sistemden tanımladığımız /burak2'i ederek çalıştırdığımızda gelecek olan save'i direkt olarak /burak yapmış olduk.
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
EXPORTING
  * I_INTERFACE_CHECK = 'X'

```

Genelde şu şekilde kullanılır;

Giriş ekranına bir tane variant parametresi tanımlanır ve aslında rapor çalışmadan kullanıcı isterse kendisinin bir tanesini seçmesine olanak sağlar.

Bunun için ise şunları yaparız;

PARAMETERS: p\_vari TYPE disvariant-variant.

```

PARAMETERS: p_vari TYPE disvariant-variant.

```

**r: Change Report ZB\_REUSE**

wsr Y ser mation System nizer ces Browser xt Browser nfo System ser vser

Select Pattern

Statements

☒ CALL FUNCTION \*REUSE\*ALV\*VARIA\*

☐ ABAP Objects Patterns

☐ MESSAGE

Class

Type

Number

☐ SELECT FROM

☐ PERFORM

☐ AUTHORITY-CHECK

Repository Info System: Function Modules Fnd (7 Hts)

Function group	Function group short text
Function Module Name	Short text for function module
SALV	SAP List Viewer
REUSE_ALV_VARIANT_ALL_MAINTAIN	Maintain all display variants for one object
REUSE_ALV_VARIANT_DEFAULT_GET	Read default display variant (description only, w/o field catalog)
REUSE_ALV_VARIANT_EXISTENCE	Checks whether a display variant exists
REUSE_ALV_VARIANT_F4	Display variant selection dialog box
REUSE_ALV_VARIANT_MAINTAIN	Definition of a display variant outside the list context
REUSE_ALV_VARIANT_SAVE	Save a display variant
REUSE_ALV_VARIANT_SELECT	Read a display variant

ZB\_REUSE

```

*&-----
*& Report ZB_REUSE
*&-----
*&
REPORT ZB_REUSE.

INCLUDE zb_reuse_top.
INCLUDE zb_reuse_frm.

INITIALIZATION.

CREATE OR REPLACE

```

"Bu da od yapısı list."

```

6 REPORT ZB_REUSE.
7
8 INCLUDE zb_reuse_top.
9 INCLUDE zb_reuse_frm.
10
11 INITIALIZATION.
12 gs_variant_get-report = sy-repid. "sy-repid ile hangi programda calistigini belirtmem gerekiyor.
13 CALL FUNCTION 'REUSE_ALV_VARIANT_DEFAULT_GET'
14     CHANGING
15         cs_variant          = gs_variant_get
16     EXCEPTIONS
17         WRONG_INPUT        = 1
18         NOT_FOUND           = 2
19         PROGRAM_ERROR       = 3
20         OTHERS              = 4
21
22 .
23 IF sy-subrc eq 0. "defaullta tanimlanmis bir sey bulabilmisse..
24     p_vari = gs_variant_get-variant. "p_vari gs_variant-variant'ını alip kendisine atiyor.
25     * Implement suitable error handling here
26 ENDIF.
27

```

INITIALIZATION.

gs\_variant\_get-report = sy-repid. "sy-repid ile hangi programda calistigini belirtmem gerekiyor.  
CALL FUNCTION 'REUSE\_ALV\_VARIANT\_DEFAULT\_GET'

CHANGING

cs\_variant = gs\_variant\_get

EXCEPTIONS

WRONG\_INPUT = 1

NOT\_FOUND = 2

PROGRAM\_ERROR = 3

OTHERS = 4

.

IF sy-subrc eq 0. "defaullta tanimlanmis bir sey bulabilmisse..

p\_vari = gs\_variant\_get-variant. "p\_vari gs\_variant-variant'ını alip kendisine atiyor.

\* Implement suitable error handling here

ENDIF.

Şimdi ise kullanıcıya yardımcı olmak için bir searchelp ataması yapıyoruz;

Select Pattern

Statements

☒ CALL FUNCTION reuse\*alv\*vari\*

☐ ABAP Objects Patterns

☐ MESSAGE

Class

Type

Number

☐ SELECT FROM

☐ PERFORM

☐ AUTHORITY-CHECK

☐ WRITE Pattern

☐ CASE for Status

Structured Data Object

☐ with fields

☐ with TYPE

Enterprise Services

☐ Service Consumer

☐ CALL DIALOG

☐ Other Pattern

Repository Info System: Function Modules Find (7 Hits)

Function group	Function group short text
Function Module Name	Short text for function module
SALV	SAP List Viewer
REUSE_ALV_VARIANT_ALL_MAINTAIN	Maintain all display variants for one object
REUSE_ALV_VARIANT_DEFAULT_GET	Read default display variant (description only, w
REUSE_ALV_VARIANT_EXISTENCE	Checks whether a display variant exists
REUSE_ALV_VARIANT_F4	Display variant selection dialog box
REUSE_ALV_VARIANT_MAINTAIN	Definition of a display variant outside the list
REUSE_ALV_VARIANT_SAVE	Save a display variant
REUSE_ALV_VARIANT_SELECT	Read a display variant

13 CALL FUNCTION 'REUSE\_ALV\_VARIANT\_DEFAULT\_GET'

14 CHANGING

15 cs\_variant = gs\_variant\_get

16 EXCEPTIONS

17 WRONG\_INPUT = 1

18 NOT\_FOUND = 2

19 PROGRAM\_ERROR = 3

20 OTHERS = 4

21

22 IF sy-subrc eq 0. "defaultta tanimlanmis bir sey bulabilmi

23 p\_vari = gs\_variant\_get-variant. "p\_vari gs\_variant-vari

24 \* Implement suitable error handling here

25 ENDIF.

26

27 at SELECTION-SCREEN on VALUE-REQUEST FOR p\_vari.

28

29

30 START-OF-SELECTION.

```
data: gs_variant type DISVARIANT,
      gs_variant_get type disvariant,
      gv_exit type char1.
```

```
at SELECTION-SCREEN on VALUE-REQUEST FOR p_vari.
CALL FUNCTION 'REUSE_ALV_VARIANT_F4'
EXPORTING
  is_variant = gs_variant_get
  I_TABNAME_HEADER =
  I_TABNAME_ITEM =
  IT_DEFAULT_FIELDCAT =
  I_SAVE = ' '
  I_DISPLAY_VIA_GRID = ' '
IMPORTING
  E_EXIT = gv_exit
  ES_VARIANT = gs_variant_get
EXCEPTIONS
  NOT_FOUND = 1
  PROGRAM_ERROR = 2
  OTHERS = 3
```

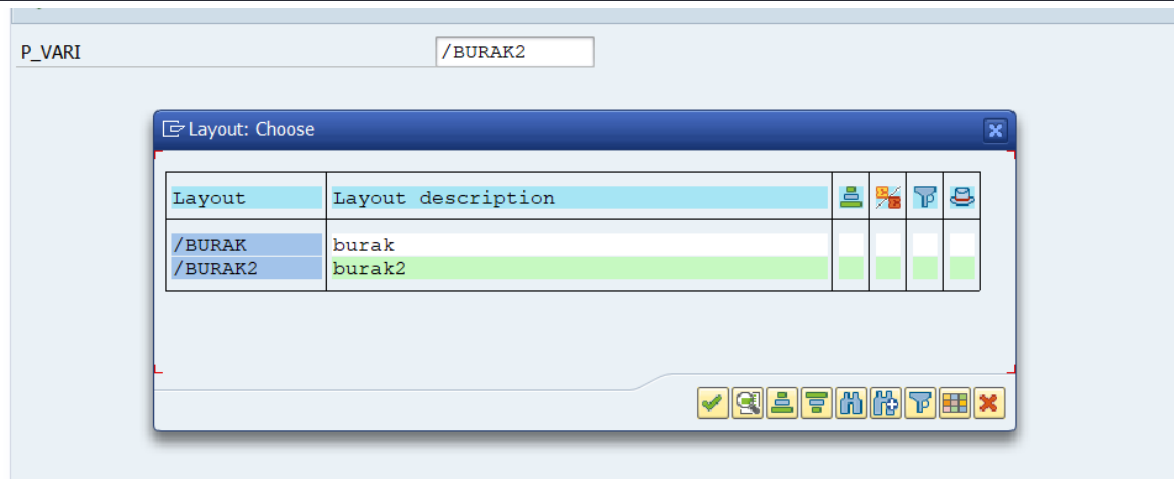
```

ENDIF sy-subrc eq 0. "defaultta tanımlanmış bir şey bulabilmişse..
  p_vari = gs_variant_get-variant. "p_vari gs_variant-variant'ını alıp kendisine atıyor.
  * Implement suitable error handling here
ENDIF.

at SELECTION-SCREEN on VALUE-REQUEST FOR p_vari.
  gs_variant_get-report = sy-repid.
  CALL FUNCTION 'REUSE_ALV_VARIANT_F4'
    EXPORTING
      is_variant          = gs_variant_get
    IMPORTING
      E_EXIT              = gv_exit
      ES_VARIANT          = gs_variant_get
    EXCEPTIONS
      NOT_FOUND           = 1
      PROGRAM_ERROR       = 2
      OTHERS              = 3
  .

IF sy-subrc eq 0.
  IF gv_exit IS INITIAL.
    p_vari = gs_variant_get-variant.
  ENDIF.
  * Implement suitable error handling here
ENDIF.

```



Son olarakta reuse içerisinde bunu tanımlamamız gerekmektedir, bunun içinde şu kısmı değiştiriyoruz;

```

gs_variant-variant = '/BURAK'. "Burada sis
CALL FUNCTION 'REUSE_ALV_VARIANT_F4'

```

**SON KULLANICI GENELDE BUNU İSTER! (VARIANT SEARCHHELP)**

## DEFAULT

```

I_DEFAULT = 'X'

```

Ön ayarlı bir varianttan etkilensin mi etkilenmesin mi? Görüldüğü haliyle X iken default hiçbir variant yokmuş gibi çalışır. Dışarıdan zorla verilen variant yapısı bunu engeller!

X gönderiyorsak ön ayarlı yapıdan etkilenir, seçili variantı onun özelliğinde getirir fakat boş verilirse alv'in kendi yapısında getirir.