**ALV (ABAP LIST VIEWER)**

**ALV (ABAP LIST VIEWER)**

SAP’ nin standart liste görüntüleyicisidir (ABAP List Viewer). İnternal tablodaki bilgileri standart liste formatında görüntüler. Standart bir liste formatı ve kullanıcı arayüzü sağlar. Hazır fonksiyonları sayesinde geliştirme zamanını önemli derecede azaltır.

Liste çıktısını oluşturmak için iki temel parametreye ihtiyaç vardır.

1. Veri tablosu: ALV’ de gösterilmek istenen internal tablodur

2. Sütun bilgileri: Sütun başlığı, genişliği gibi bilgilerdir.

ALV birçok hazır fonksiyona sahiptir. Bazı hazır fonksiyonlar,

* Sıralama
* Filtreleme
* Alt toplam ve tra toplam alma
* Raporu çeşitli formatlarda bilgisayara kaydetme
* Sütunların sırasını değiştirme, istenmeyen sütunları gizleme ve variant olarak sütun sırasının kaydedilmesi

ALV fonksiyonlarının olduğu araç çubuğu.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755710e0afe.png

ALV çıktısı klasik veya nesne tabanlı (OO) alv foksiyonlar kullanılabilir.

Klasik ALV raporlarında kullanılacak bazı fonksiyonlar aşağıdaki tablodadır.

|  |  |
| --- | --- |
| **Fonksiyon Modül** | **Tanım** |
| REUSE\_ALV\_GRID\_DISPLAY | Veriyi ızgara (grid) içerisinde gösteren fonksiyon |
| REUSE\_ALV\_LIST\_DISPLAY | Veriyi standart liste çıktısı olarak gösteren fonksiyon |
| REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY | Veriyi hiyerarşik sıralı liste içerisinde gösteren fonksiyon |
| REUSE\_ALV\_POPUP\_TO\_SELECT | Popup pencerede bir listeden seçim yapılmasını sağlayan fonksiyon |
| REUSE\_ALV\_COMMENTARY\_WRITE | ALV’ ye başlık alanı ekler |
| REUSE\_ALV\_VARIANT\_F4 | Seçim ekranında variant seçimi yapılmasını sağlar |
| REUSE\_ALV\_VARIANT\_DEFAULT\_GET | Varsayılan görüntü varyantını getirir |
| REUSE\_ALV\_VARIANT\_EXISTENCE | Bir variant’ ın üst birimde bulunup bulunmadığını kontrol eder |
| REUSE\_ALV\_FIELDCATALOG\_MERGE | Abap dictionary’ deki nesnelerden veya internal tablodan alan kataloğu oluşturmak için kullanılır |
| REUSE\_ALV\_EVENTS\_GET | ALV için kullanılabilecek olayların (events) listesini verir |

**REUSE\_ALV\_GRID\_DISLAY**

Parametre olarak verilerin internal tabloyu hücresel formatta görüntüler.

Sütun bilgilerini aktarmak için “I\_STRUCTURE\_NAME” parametresi veya “IT\_FIELDCAT” parametresi kullanılır.

I\_STRUCTURE\_NAME: Abap dictionary’ deki tanımlı bir yapıdır. Structure, tablo, tablo tipi gibi. Bu parametre kullanıldığında yapıdaki tüm sütunlar ALV’ de gösterilmeye çalışılır.

IT\_FIELDCAT: Sütun bilgilerini içeren internal tabloyu (field catalog) parametre olarak alır.

Alan kataloğu internal tablosu üç yönetim ile hazırlanabilir.

1. Alan kataloğu internal tablosunu el ile oluşturarak

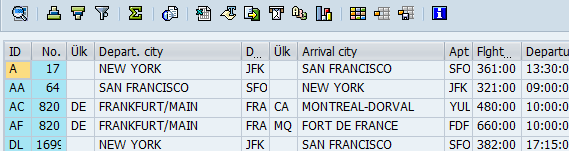
2. Alan kataloğu internal tablosunu fonksiyon ile Abap Dictionary’ daki structure, tablo veya tablo tiplerinden oluşturarak

3. Önce Abap dictionary’ deki nesnelerden oluşturulur, daha sonra istenilen şekilde düzeltilerek

|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| I\_INTERFACE\_CHECK |  |  | SPACE | X |  | Arayüz tutarlılık denetimi. Liste görünümünde &SOS komutu ile de çalışır. |
| I\_BYPASSING\_BUFFER | TYPE | CHAR01 | SPACE | X | X | Bufferları (tampon) göz ardı et |
| I\_BUFFER\_ACTIVE |  |  | SPACE | X | X | Buffer (tamponlama) aktif |
| I\_CALLBACK\_PROGRAM | LIKE | SY-REPID | SPACE | X |  | Fonksiyonu çağıran program ismi |
| I\_CALLBACK\_PF\_STATUS\_SET | TYPE | SLIS\_FORMNAME | SPACE | X |  | GUI status alt yordam ismi |
| I\_CALLBACK\_USER\_COMMAND | TYPE | SLIS\_FORMNAME | SPACE | X |  | Kullanıcı komutu yakalamak için alt yordam ismi |
| I\_CALLBACK\_TOP\_OF\_PAGE | TYPE | SLIS\_FORMNAME | SPACE | X |  | TOP-OF-PAGE için alt yordam ismi |
| I\_CALLBACK\_HTML\_TOP\_OF\_PAGE | TYPE | SLIS\_FORMNAME | SPACE | X |  | HTML TOP-OF-PAGE için alt yordam ismi |
| I\_CALLBACK\_HTML\_END\_OF\_LIST | TYPE | SLIS\_FORMNAME | SPACE | X |  | HTML END-OF-LIST için alt yordam ismi |
| I\_STRUCTURE\_NAME | LIKE | DD02L-TABNAME |  | X |  | Abap dictionary içerisindeki structure |
| I\_BACKGROUND\_ID | TYPE | SDYDO\_KEY | SPACE | X |  | Top-of-Page alanındaki çıktı için arkaplan parametresi |
| I\_GRID\_TITLE | TYPE | LVC\_TITLE |  | X |  | ALV’nin başlık metni |
| I\_GRID\_SETTINGS | TYPE | LVC\_S\_GLAY |  | X |  | Top-of-Page ve End-of-List görünümleri için görünüm ayarları |
| IS\_LAYOUT | TYPE | SLIS\_LAYOUT\_ALV |  | X |  | Listenin görünüm düzeni |
| IT\_FIELDCAT | TYPE | SLIS\_T\_FIELDCAT\_ALV |  | X |  | Sütun bilgilerini içeren alan kataloğu (field catalog) internal tablosu |
| IT\_EXCLUDING | TYPE | SLIS\_T\_EXTAB |  | X |  | Görünmemesi istenen ALV fonksiyonlarını aktarmak için kullanılan internal tablo |
| IT\_SPECIAL\_GROUPS | TYPE | SLIS\_T\_SP\_GROUP\_ALV |  | X |  | Sütun seçimi için alanların gruplama bilgisi |
| IT\_SORT | TYPE | SLIS\_T\_SORTINFO\_ALV |  | X |  | Çıktı için sıralama bilgisi |
| IT\_FILTER | TYPE | SLIS\_T\_FILTER\_ALV |  | X |  | Çıktı için filtreleme bilgisi |
| IS\_SEL\_HIDE | TYPE | SLIS\_SEL\_HIDE\_ALV |  | X |  | Artık desteklenmiyor |
| I\_DEFAULT |  |  | ‘X’ | X |  | Başlangıç variant seçimini aktif veya pasif yapar |
| I\_SAVE |  |  | SPACE | X |  | Aldığı parametrelere göre variant’ ın saklanmasına izin verir. |
| IS\_VARIANT | LIKE | DISVARIANT |  |  |  | Variant saklanacak ise kullanılır. Variant bilgilesi aktarılır |
| IT\_EVENTS | TYPE | SLIS\_T\_EVENT |  | X |  | Kullanılacak olayların listes |
| IT\_EVENT\_EXIT | TYPE | SLIS\_T\_EVENT\_EXIT |  | X |  | Standart ALV fonksiyonları internal tablosu. Burada belirtilen fonksiyonlar USER\_COMMAND fonksiyonuna düşmesi |
| IS\_PRINT | TYPE | SLIS\_PRINT\_ALV |  | X |  | Çıktı bilgisi |
| IS\_REPREP\_ID | TYPE | SLIS\_REPREP\_ID |  | X |  | Arayüz başlangıç anahtarı |
| I\_SCREEN\_START\_COLUMN |  |  | 0 | X |  | Listenin sütun başlangıç pozisyonu. Sadece dialog liste’ te görünecek ise kullanılır |
| I\_SCREEN\_START\_LINE |  |  | 0 | X |  | Listenin satır başlangıç pozisyonu. Sadece dialog liste’ te görünecek ise kullanılır |
| I\_SCREEN\_END\_COLUMN |  |  | 0 | X |  | Listenin sütun bitiş pozisyonu. Sadece dialog liste’ te görünecek ise kullanılır |
| I\_SCREEN\_END\_LINE |  |  | 0 | X |  | Listenin satır bitiş pozisyonu. Sadece dialog liste’ te görünecek ise kullanılır |
| I\_HTML\_HEIGHT\_TOP | TYPE | I | 0 | X |  | HTML\_TOP\_OF\_PAGE yüksekliği |
| I\_HTML\_HEIGHT\_END | TYPE | I | 0 | X |  | HTML\_END\_OF\_PAGE yüksekliği |
| IT\_ALV\_GRAPHICS | TYPE | DTC\_T\_TC |  | X |  | ALV grafiği için parametre |
| IT\_HYPERLINK | TYPE | LVC\_T\_HYPE |  | X |  | Hiperlink |
| IT\_ADD\_FIELDCAT | TYPE | SLIS\_T\_ADD\_FIELDCAT |  | X |  | Ek alan kataloğu bilgileri |
| IT\_EXCEPT\_QINFO | TYPE | SLIS\_T\_QINFO\_ALV |  | X |  |  |
| IR\_SALV\_FULLSCREEN\_ADAPTER | TYPE REF TO | CL\_SALV\_FULLSCREEN\_ADAPTER |  | X |  | Kullanılamaz |
| **EXPORT** |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Dğr atn.** | **Parametre açıklaması** |  |  |
| E\_EXIT\_CAUSED\_BY\_CALLER |  |  |  | CALLBACK\_USER\_COMMAND çıkış parametresi |  |  |
| ES\_EXIT\_CAUSED\_BY\_USER | TYPE | SLIS\_EXIT\_BY\_USER |  | Kullanıcının ALV’ den çıkarken hangi butonu kullandığı bilgisi |  |  |
| **TABLES** |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Dğr atn.** | **Parametre açıklaması** |  |  |
| T\_OUTTAB |  |  |  | ALV’ de gösterilecek veri |  |  |
| **EXCEPTIONS** |  |  |  |  |  |  |
| **Exception** | **Tanım** |  |  |  |  |  |
| PROGRAM\_ERROR | Program hatası |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

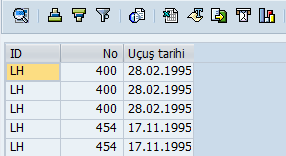
Örnek 1: “I\_STRUCTURE\_NAME” parametresi kullanılan örnek. Abap dictionary’ deki yapı ALV’ nin sütunlarını oluşturmak için kullanılır.

DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gv\_repid TYPE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF  TABLE gt\_spfli.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
i\_structure\_name    = ‘SPFLI’  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_spfli.

  
Örnek 2: “IT\_FIELDCAT” parametresi kullanılan örnek. Sütun bilgileri, satır satır internal tabloya ekleniyor ve fonksiyona veriliyor.

TYPE-POOLS: slis.

DATA: gt\_fieldcat  TYPE slis\_t\_fieldcat\_alv,  
gs\_fieldcat  TYPE slis\_fieldcat\_alv.  
Veri tablosu tipi tanımlanıyor  
DATA  gt\_sbook TYPE STANDARD TABLE OF sbook.  
DATA gv\_repid TYPE sy-repid.  
Veri tablosu dolduruluyor  
SELECT \* FROM sbook INTO CORRESPONDING FIELDS OF TABLE gt\_sbook.  
Alan kataloğu dolduruluyor  
gs\_fieldcat-fieldname  = ‘CARRID’.  
gs\_fieldcat-seltext\_m  = ‘ID’.  
gs\_fieldcat-outputlen  = 7.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gs\_fieldcat-fieldname  = ‘CONNID’.  
gs\_fieldcat-seltext\_m  = ‘No’.  
gs\_fieldcat-outputlen  = 8.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gs\_fieldcat-fieldname  = ‘FLDATE’.  
gs\_fieldcat-seltext\_m  = ‘Uçuş tarihi’.  
gs\_fieldcat-outputlen  = 9.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
it\_fieldcat         = gt\_fieldcat  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_sbook  
EXCEPTIONS  
program\_error       = 1  
OTHERS              = 2.

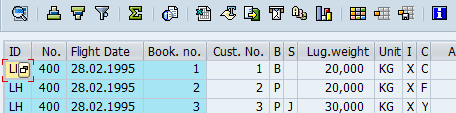


Açıklama: Structure’ a atanan aşağıdaki değerler “slis\_t\_fieldcat\_alv” tablo tipindeki internal tabloya eklenir. İnternal tabloya eklenen her bir satır ALV’ de bir sütuna karşılık gelir.

* (fieldname): Veri tablosunda gösterilecek alan
* (seltext\_m): Alanın sütun başlığı
* (outputlen) :Alanın sütun genişliği

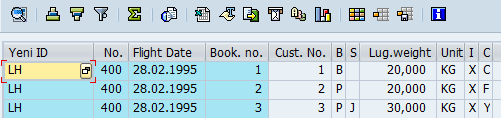
Örnek 3: Abap dictionary’ deki bir yapıdaki alan bilgilerinden sütun bilgileri oluşturuluyor. Sütun başlığı, genişliği gibi bilgiler yapıdaki alanların özelliklerinden (data element ve domain) oluşturulur.

TYPE-POOLS: slis.  
DATA : gt\_sbook TYPE STANDARD TABLE OF sbook,  
gv\_repid TYPE sy-repid.  
DATA  gt\_fieldcat  TYPE slis\_t\_fieldcat\_alv.  
SELECT \* FROM sbook INTO CORRESPONDING FIELDS OF TABLE gt\_sbook.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_FIELDCATALOG\_MERGE’  
EXPORTING  
i\_program\_name           = gv\_repid  
i\_structure\_name         = ‘SBOOK’  
CHANGING  
ct\_fieldcat             = gt\_fieldcat  
EXCEPTIONS  
inconsistent\_interface   = 1  
program\_error            = 2  
OTHERS                   = 3.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
it\_fieldcat         = gt\_fieldcat  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_sbook  
EXCEPTIONS  
program\_error       = 1  
OTHERS              = 2.



Örnek 4: “REUSE\_ALV\_FIELDCATALOG\_MERGE” fonksiyonu kullanılarak sütun bilgilerini Abap dictionary’ deki bir yapıdan alan ve daha sonra sütun bilgilerinde değişiklik yapan örnek.

TYPE-POOLS: slis.  
\* Alan kataloğu tanımlanıyor  
DATA: gt\_fieldcat  TYPE slis\_t\_fieldcat\_alv,  
gs\_fieldcat  TYPE slis\_fieldcat\_alv.  
\* Veri tablosu tipi tanımlanıyor  
DATA  gt\_sbook     TYPE STANDARD TABLE OF sbook.  
DATA gv\_repid TYPE sy-repid.  
\* Veri tablosu dolduruluyor  
SELECT \* FROM sbook INTO CORRESPONDING FIELDS OF TABLE gt\_sbook.  
\* Alan kataloğu internal tablodan oluşturuluyor  
CALL FUNCTION ‘REUSE\_ALV\_FIELDCATALOG\_MERGE’  
EXPORTING  
i\_program\_name           = gv\_repid  
i\_structure\_name         = ‘SBOOK’  
CHANGING  
ct\_fieldcat             = gt\_fieldcat  
EXCEPTIONS  
inconsistent\_interface   = 1  
program\_error            = 2  
OTHERS                   = 3.  
\* Otomatik oluşturulan alan kataloğunu üzerinde değişiklik  
LOOP AT gt\_fieldcat INTO gs\_fieldcat WHERE fieldname = ‘CARRID’.  
gs\_fieldcat-seltext\_m  = ‘Yeni ID’.  
gs\_fieldcat-outputlen  = 12.  
MODIFY gt\_fieldcat FROM gs\_fieldcat.  
ENDLOOP.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
it\_fieldcat         = gt\_fieldcat  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_sbook  
EXCEPTIONS  
program\_error       = 1  
OTHERS              = 2.



**REUSE\_ALV\_LIST\_DISPLAY**

Parametre olarak verilerin internal tabloyu liste formatında görüntüler. Kullanım olarak “REUSE\_ALV\_GRID\_DISPLAY” fonksiyonuna çok benzer, aldığı parametre sayısı daha azdır.

Sütun bilgilerini aktarmak için “I\_STRUCTURE\_NAME” parametresi veya “IT\_FIELDCAT” parametresi kullanılır.

I\_STRUCTURE\_NAME: Abap dictionary’ deki tanımlı bir yapıdır. Structure, tablo, tablo tipi gibi. Bu parametre kullanıldığında yapıdaki tüm sütunlar ALV’ de gösterilmeye çalışılır.

IT\_FIELDCAT: Sütun bilgilerini içeren internal tabloyu (field catalog) parametre olarak alır.

Alan kataloğu internal tablosu üç yönetim ile hazırlanabilir.

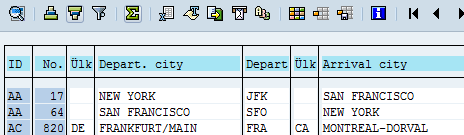
1. Alan kataloğu internal tablosunu el ile oluşturarak

2. Alan kataloğu internal tablosunu fonksiyon ile Abap Dictionary’ daki structure, tablo veya tablo tiplerinden oluşturarak

3. Önce Abap dictionary’ deki nesnelerden oluşturulur, daha sonra istenilen şekilde düzeltilerek

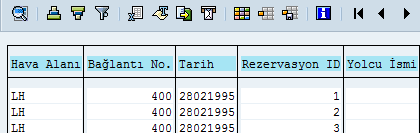
Örnek 1: “I\_STRUCTURE\_NAME” parametresi kullanılan örnek. Abap dictionary’ deki yapı ALV’ nin sütunlarını oluşturmak için kullanılır.

DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gv\_repid TYPE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_LIST\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
i\_structure\_name    = ‘SPFLI’  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_spfli.



Örnek 2: “IT\_FIELDCAT” parametresi kullanılan örnek. Sütun bilgileri, satır satır internal tabloya ekleniyor ve fonksiyona veriliyor.

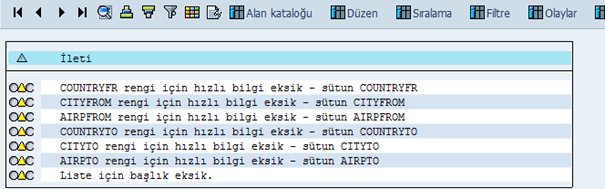
TYPE-POOLS: slis.  
\* Alan kataloğu tanımlanıyor  
DATA: gt\_fieldcat  TYPE slis\_t\_fieldcat\_alv,  
gs\_fieldcat  TYPE slis\_fieldcat\_alv.  
\* Veri tablosu tip tanımlanıyor  
DATA: gt\_sbook     TYPE STANDARD TABLE OF sbook.  
DATA gv\_repid TYPE sy-repid.  
\* Veri tablosu dolduruluyor  
SELECT \* FROM sbook INTO CORRESPONDING FIELDS OF TABLE gt\_sbook.  
\* Alan kataloğu dolduruluyor  
gs\_fieldcat-fieldname  = ‘CARRID’.  
gs\_fieldcat-seltext\_m  = ‘Hava Alanı’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gs\_fieldcat-fieldname  = ‘CONNID’.  
gs\_fieldcat-seltext\_m  = ‘Bağlantı No.’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gs\_fieldcat-fieldname  = ‘FLDATE’.  
gs\_fieldcat-seltext\_m  = ‘Tarih’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gs\_fieldcat-fieldname  = ‘BOOKID’.  
gs\_fieldcat-seltext\_m  = ‘Rezervasyon ID’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gs\_fieldcat-fieldname  = ‘PASSNAME’.  
gs\_fieldcat-seltext\_m  = ‘Yolcu İsmi’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_LIST\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
it\_fieldcat         = gt\_fieldcat  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_sbook  
EXCEPTIONS  
program\_error       = 1  
OTHERS              = 2.



Örnek 3: “I\_INTERFACE\_CHECK” parametresi kullanılarak çalıştırılan ALV örneği. Bu parametre geliştirme ve test amaçlı kullanım için uygundur.

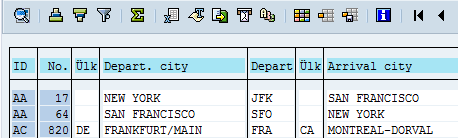
DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gv\_repid TYPE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_LIST\_DISPLAY’  
EXPORTING  
i\_interface\_check   = ‘X’  
i\_callback\_program  = gv\_repid  
i\_structure\_name    = ‘SPFLI’  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_spfli.

Program çalıştığında Alan kataloğu ile ilgili bilgieri görüntüler. Araç çubuğundan istenilen internal tablo görüntülenebilir.



Liste çıktısını görebilmek için geriye gitmek veya araç çubuğundan “Çıktı tablosu” butonuna tıklamak gerekir.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_527558e38ec7c.png



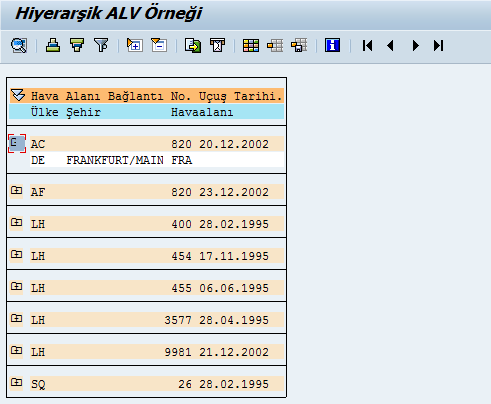
**REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY**

Hiyerarşik sıralı çıktı çıktı oluşturur. İki tane internal tabloyu formatlı ve sıralı biçimde görüntüler. Aldığı parametrelerin çoğu “REUSE\_ALV\_GRID\_DISPLAY” ve “REUSE\_ALV\_LIST\_DISPLAY” fonksiyonu ile aynıdır. Ek olarak detay olarak gösterilecek internal tablo bilgileri ve iki tablo arasındaki anahtar alanlar parametre olarak gönderilir.

|  |  |  |
| --- | --- | --- |
| **Parametre** | **Açıklama** | **Fonksiyon** |
| I\_TABNAME\_HEADER | Başlık olarak gösterilecek internal tablo ismi | IMPORT |
| I\_TABNAME\_ITEM | Detay olarak gösterilecek internal tablo ismi | IMPORT |
| IS\_KEYINFO | İki tablo arasındaki anahtar alan bağlantısı | IMPORT |
| T\_OUTTAB\_HEADER | Başlık olarak gösterilecek internal tablo | TABLES |
| T\_OUTTAB\_ITEM | Detay olarak gösterilecek internal tablo | TABLES |

Örnek: SFLIGHT tablosunu başlık SPFLI tablosunun detay olarak gösterildiği hiyerarşik ALV örneği.

TYPE-POOLS: slis.  
DATA: gt\_fieldcat\_spfli   TYPE slis\_t\_fieldcat\_alv,  
gt\_fieldcat\_sflight TYPE slis\_t\_fieldcat\_alv.  
TYPES: BEGIN OF gy\_sflight,  
carrid TYPE sflight-carrid,  
connid TYPE sflight-connid,  
fldate TYPE sflight-fldate,  
expand(1) TYPE c,  
END OF gy\_sflight.  
DATA: gt\_sflight TYPE STANDARD TABLE OF gy\_sflight,  
gt\_spfli TYPE STANDARD TABLE OF spfli.  
START-OF-SELECTION.  
PERFORM tablolari\_doldur.  
PERFORM alankatalogu\_olustur.  
PERFORM tablolari\_goster.  
FORM tablolari\_doldur .  
SELECT \* FROM sflight  
INTO CORRESPONDING FIELDS OF TABLE gt\_sflight.  
SELECT \* FROM spfli  
INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
ENDFORM.              ” TABLOLARI\_DOLDUR  
FORM alankatalogu\_olustur .  
DATA : gs\_fieldcat  TYPE slis\_fieldcat\_alv.  
gs\_fieldcat-col\_pos    = 1.  
gs\_fieldcat-fieldname  = ‘CARRID’.  
gs\_fieldcat-tabname    = ‘GT\_SFLIGHT’.  
gs\_fieldcat-seltext\_m  = ‘Hava Alanı’.  
APPEND gs\_fieldcat TO gt\_fieldcat\_sflight.  
gs\_fieldcat-col\_pos    = 2.  
gs\_fieldcat-fieldname  = ‘CONNID’.  
gs\_fieldcat-tabname    = ‘GT\_SFLIGHT’.  
gs\_fieldcat-seltext\_m  = ‘Bağlantı No.’.  
APPEND gs\_fieldcat TO gt\_fieldcat\_sflight.  
gs\_fieldcat-col\_pos    = 3.  
gs\_fieldcat-fieldname  = ‘FLDATE’.  
gs\_fieldcat-tabname    = ‘GT\_SFLIGHT’.  
gs\_fieldcat-seltext\_m  = ‘Uçuş Tarihi.’.  
APPEND gs\_fieldcat TO gt\_fieldcat\_sflight.  
gs\_fieldcat-fieldname  = ‘COUNTRYFR’.  
gs\_fieldcat-tabname    = ‘GT\_SPFLI’.  
gs\_fieldcat-seltext\_m  = ‘Ülke’.  
APPEND gs\_fieldcat TO gt\_fieldcat\_sflight.  
gs\_fieldcat-fieldname  = ‘CITYFROM’.  
gs\_fieldcat-tabname    = ‘GT\_SPFLI’.  
gs\_fieldcat-seltext\_m  = ‘Şehir’.  
APPEND gs\_fieldcat TO gt\_fieldcat\_sflight.  
gs\_fieldcat-fieldname  = ‘AIRPFROM’.  
gs\_fieldcat-tabname    = ‘GT\_SPFLI’.  
gs\_fieldcat-seltext\_m  = ‘Havaalanı’.  
APPEND gs\_fieldcat TO gt\_fieldcat\_sflight.  
ENDFORM.                    ” ALANKATALOGU\_OLUSTUR  
FORM tablolari\_goster .  
DATA: lv\_tabname\_header  TYPE slis\_tabname,  
lv\_tabname\_item    TYPE slis\_tabname,  
lv\_keyinfo         TYPE slis\_keyinfo\_alv,  
lv\_repid           TYPE sy-repid,  
ls\_layout          TYPE slis\_layout\_alv.  
lv\_repid = sy-repid.  
ls\_layout-expand\_fieldname  = ‘EXPAND’.  
ls\_layout-window\_titlebar   = ‘Hiyerarşik ALV Örneği’.  
ls\_layout-colwidth\_optimize = ‘X’.  
lv\_tabname\_header    = ‘GT\_SFLIGHT’.  
lv\_tabname\_item      = ‘GT\_SPFLI’.  
lv\_keyinfo-header01  = ‘CARRID’.  
lv\_keyinfo-item01    = ‘CARRID’.  
lv\_keyinfo-header02  = ‘CONNID’.  
lv\_keyinfo-item02    = ‘CONNID’.  
CALL FUNCTION ‘REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY’  
EXPORTING  
i\_callback\_program      = lv\_repid  
is\_layout               = ls\_layout  
it\_fieldcat             = gt\_fieldcat\_sflight  
i\_save                  = ‘A’  
i\_tabname\_header        = lv\_tabname\_header  
i\_tabname\_item          = lv\_tabname\_item  
is\_keyinfo              = lv\_keyinfo  
TABLES  
t\_outtab\_header         = gt\_sflight  
t\_outtab\_item           = gt\_spfli.  
ENDFORM.                    <I>” TABLOLARI\_GOSTER</I>



**REUSE\_ALV\_BLOCK\_LIST\_APPEND**

Birden fazla Listeyi ard arda görüntüler. Veri tablosu ve alan kataloğu oluşturulan listeler sırasıyla eklenir. Daha sonra “REUSE\_ALV\_BLOCK\_LIST\_DISPLAY” fonksiyonu ile eklenen listeler görüntülenir.

Aşağıdaki parametreler zorunludur.

· IS\_LAYOUT

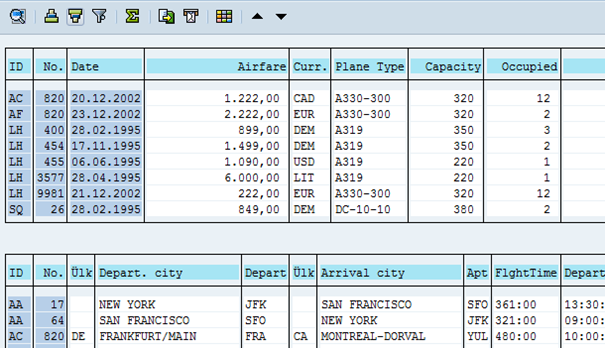
· IT\_FIELDCAT

· I\_TABNAME

· IT\_EVENTS

Örnek: Ard arda iki tane ayrı listenin eklendiği örnek.

TYPE-POOLS: slis.  
DATA: gt\_fieldcat\_sflight TYPE slis\_t\_fieldcat\_alv,  
gt\_fieldcat\_spfli TYPE slis\_t\_fieldcat\_alv.  
DATA: gt\_sflight TYPE STANDARD TABLE OF sflight,  
gt\_spfli TYPE STANDARD TABLE OF spfli.  
START-OF-SELECTION.  
PERFORM tablolari\_doldur.  
PERFORM alankatalogu\_olustur.  
PERFORM tablolari\_goster.  
FORM tablolari\_doldur .  
SELECT \* FROM sflight INTO CORRESPONDING FIELDS OF TABLE gt\_sflight.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
ENDFORM.                    <I>” TABLOLARI\_DOLDUR</I>  
FORM alankatalogu\_olustur .  
CALL FUNCTION ‘REUSE\_ALV\_FIELDCATALOG\_MERGE’  
EXPORTING  
i\_structure\_name             = ‘SFLIGHT’  
CHANGING  
ct\_fieldcat                 = gt\_fieldcat\_sflight  
EXCEPTIONS  
inconsistent\_interface       = 1  
program\_error                = 2  
OTHERS                       = 3.  
CALL FUNCTION ‘REUSE\_ALV\_FIELDCATALOG\_MERGE’  
EXPORTING  
i\_structure\_name             = ‘SPFLI’  
CHANGING  
ct\_fieldcat                 = gt\_fieldcat\_spfli  
EXCEPTIONS  
inconsistent\_interface       = 1  
program\_error                = 2  
OTHERS                       = 3.  
ENDFORM.                    ” ALANKATALOGU\_OLUSTUR  
FORM tablolari\_goster .  
DATA : lv\_repid TYPE sy-repid,  
ls\_layout TYPE slis\_layout\_alv,  
lt\_events TYPE slis\_t\_event.  
lv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_BLOCK\_LIST\_INIT’  
EXPORTING  
i\_callback\_program          = lv\_repid.  
CALL FUNCTION ‘REUSE\_ALV\_BLOCK\_LIST\_APPEND’  
EXPORTING  
is\_layout                   = ls\_layout  
it\_fieldcat                 = gt\_fieldcat\_sflight  
i\_tabname                   = ‘GT\_SFLIGHT’  
it\_events                   = lt\_events  
TABLES  
t\_outtab                   = gt\_sflight  
EXCEPTIONS  
program\_error               = 1  
maximum\_of\_appends\_reached  = 2  
OTHERS                        = 3.  
CALL FUNCTION ‘REUSE\_ALV\_BLOCK\_LIST\_APPEND’  
EXPORTING  
is\_layout                   = ls\_layout  
it\_fieldcat                 = gt\_fieldcat\_spfli  
i\_tabname                   = ‘GT\_SPFLI’  
it\_events                   = lt\_events  
TABLES  
t\_outtab                   = gt\_spfli  
EXCEPTIONS  
program\_error               = 1  
maximum\_of\_appends\_reached  = 2  
OTHERS                      = 3.  
CALL FUNCTION ‘REUSE\_ALV\_BLOCK\_LIST\_DISPLAY’  
EXCEPTIONS  
program\_error                 = 1  
OTHERS                        = 2.  
IF sy-subrc <> 0.  
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
ENDIF.  
ENDFORM.                    ” TABLOLARI\_GOSTER



**REUSE\_ALV\_FIELDCATALOG\_MERGE**

Alan kataloğu (ALV’ deki sütun bilgileri) internal tablosunu otomatik olarak oluşturur. Fonksiyona gönderilen “I\_INTERNAL\_TABNAME” veya “I\_STRUCTURE\_NAME” parametresine göre alan kataloğu internal tablosunu doldurur.

“I\_INTERNAL\_TABNAME”: Program içerisinden aktarılan internal tabloya göre alan kataloğu tablosu doldurulur.

“I\_STRUCTURE\_NAME”: Abap dictionary içerisinde tanımlı, structure, tablo veya view kullanarak tabloyu doldurur.

|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** |  | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| I\_PROGRAM\_NAME | LIKE | SY-REPID |  |  | X | X | Fonksiyonu çağıran program |
| I\_INTERNAL\_TABNAME | TYPE | SLIS\_TABNAME |  |  | X | X | Alan kataloğu oluşturulacak internal tablo |
| I\_STRUCTURE\_NAME | LIKE | DD02L-TABNAME |  |  | X | X | Alan kataloğu oluşturulacak Abap dictionary nenesi |
| I\_CLIENT\_NEVER\_DISPLAY | TYPE | SLIS\_CHAR\_1 |  | X | X | X | Üst birim alanlarını gizle |
| I\_INCLNAME | LIKE | TRDIR-NAME |  |  | X | X | Include ismie |
| I\_BYPASSING\_BUFFER | TYPE | CHAR01 |  |  | X | X | Tampon belleği kullanma |
| I\_BUFFER\_ACTIVE | TYPE | CHAR01 |  |  | X | X | Özel tamponu aktif et |
| **CHANGING** |  |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |  |  |
| CT\_FIELDCAT | TYPE | SLIS\_T\_FIELDCAT\_ALV |  | X | Alan kataloğu tablosu |  |  |
| **EXCEPTIONS** |  |  |  |  |  |  |  |
| **Exception** | **Tanım** |  |  |  |  |  |  |
| INCONSISTENT\_INTERFACE | Çalıştırma parametreleri hatası |  |  |  |  |  |  |
| PROGRAM\_ERROR | Program hatası |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

“I\_STRUCTURE\_NAME” ile ilgili örnekler için “REUSE\_ALV\_GRID\_DISPLAY” fonksiyonunun anlatıldığı 2. Ve 3. Örnekleri inceleyebilirsiniz.

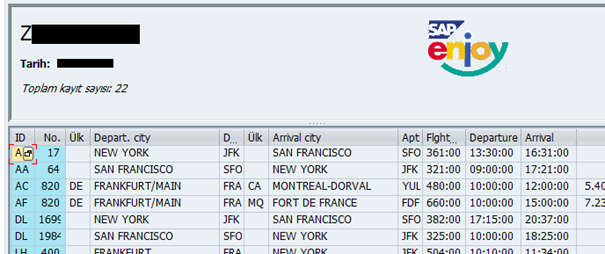
**REUSE\_ALV\_COMMENTARY\_WRITE**

ALV’ ye açıklama bilgisi ekler. Raporla ilgili bilgileri başlıkta açıklama alanında göstermek için kullanılır. Eklenmek istenen bilgiler satır satır internal tabloya eklenir ve fonksiyona parametre olarak gönderilir. “REUSE\_ALV\_GRID\_DISPLAY” fonksiyonundan “I\_CALLBACK\_TOP\_OF\_PAGE” parametresine başlık fonksiyonun kullanıldığı alt yordam ismi yazılır.

|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| IT\_LIST\_COMMENTARY | TYPE | SLIS\_T\_LISTHEADER |  |  | X | Açıklama bilgisinin olduğu internal tablo |
| I\_LOGO |  |  |  | X |  | Eklenecek logo ismi |
| I\_END\_OF\_LIST\_GRID |  |  |  | X |  |  |
| I\_ALV\_FORM | TYPE | SAP\_BOOL |  | X |  |  |

Örnek: Açıklama alanına rapor ismi, tarih, kayıt sayısı ve logo ekleyen örnek.

TYPE-POOLS: slis.  
DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gv\_repid LIKE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
\* Program ismi gv\_repid değikenine atanıyor  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program      = gv\_repid  
i\_structure\_name        = ‘SPFLI’  
i\_callback\_top\_of\_page  = ‘TOP\_OF\_PAGE’  
i\_save                  = ‘A’  
TABLES  
t\_outtab                = gt\_spfli.  
FORM top\_of\_page.  
DATA: lt\_header    TYPE slis\_t\_listheader,  
ls\_header    TYPE slis\_listheader,  
lv\_lines     TYPE i,  
lv\_linesc(4) TYPE c.  
\* Başlık alanına rapor ismi yazılıyor  
ls\_header-typ  = ‘H’.  
ls\_header-key  = ”.  
ls\_header-info = gv\_repid.  
APPEND ls\_header TO lt\_header.  
\* Günün tarihi ekleniyor  
CLEAR ls\_header.  
ls\_header-typ = ‘S’.  
ls\_header-key = ‘Tarih: ‘.  
WRITE sy-datum TO ls\_header-info.  
APPEND ls\_header TO lt\_header.  
CLEAR: ls\_header.  
\* İnternal tablodaki kayıt sayısı ekleniyor  
CLEAR ls\_header.  
\* İnternal tablodaki kayıt sayısı lv\_lines değişkenine atanıyor  
DESCRIBE TABLE gt\_spfli LINES lv\_lines.  
lv\_linesc = lv\_lines.  
CONCATENATE ‘Toplam kayıt sayısı: ‘ lv\_linesc INTO ls\_header-info SEPARATED BY space.  
ls\_header-typ  = ‘A’.  
APPEND ls\_header TO lt\_header.  
CALL FUNCTION ‘REUSE\_ALV\_COMMENTARY\_WRITE’  
EXPORTING  
it\_list\_commentary   = lt\_header  
i\_logo               = ‘ENJOYSAP\_LOGO’.  
ENDFORM.



**REUSE\_ALV\_VARIANT\_F4**

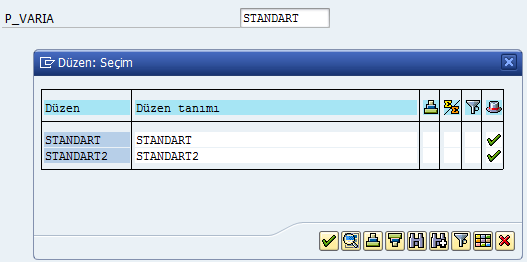
ALV variantının seçilmesini seçilmesini sağlayan bir pencere (dialog) açar. Variant listesinin hali hazırda kaydedilmiş variant/variantların olması gerekir. Çalıştırılmadan önce “IS\_VARIANT” import paremetresindeki “REPORT” değişkenine raporun ismi atanmalıdır. Pencereden seçim yapıldıktan sonra ALV fonksiyonuna seçilen ALV bilgileri yine “IS\_VARIANT” parametresi ile aktarılır.

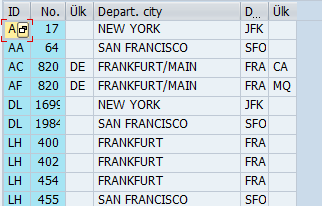
|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| IS\_VARIANT | LIKE | DISVARIANT |  |  | X | Variant bilgileri |
| I\_TABNAME\_HEADER | TYPE | SLIS\_TABNAME |  | X | X | Tablo ismi balığı |
| I\_TABNAME\_ITEM | TYPE | SLIS\_TABNAME |  | X | X | Tablo ismi kalemi |
| IT\_DEFAULT\_FIELDCAT | TYPE |  |  | X | X | Alan kataloğu internal tablosu |
| I\_SAVE |  | SLIS\_T\_FIELDCAT\_ALV | SPACE | X | X | Variant saklama seçeneği |
| I\_DISPLAY\_VIA\_GRID | TYPE | CHAR01 | SPACE | X | X | Izgara biçiminde göster |
| **EXPORT** |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Dğr atn.** | **Parametre açıklaması** |  |  |
| E\_EXIT |  |  | X | İptal seçildi |  |  |
| ES\_VARIANT | LIKE | DISVARIANT | X | Variant bilgisi |  |  |
| **EXCEPTIONS** |  |  |  |  |  |  |
| **Exception** | **Tanım** |  |  |  |  |  |
| NOT\_FOUND | Variant bulunamadı |  |  |  |  |  |
| PROGRAM\_ERROR | Program hatası |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

Örnek 1: Seçim ekranında variant seçim parametresi olan ve bu parametreden seçilen varainta göre ALV’ nin düzeninin değiştiği örnek. Variant listesini seçim ekranında görmek için önce ALV listesinde variant kaydedilmesi gerekir. Kaydedilen variant daha sonra rapor her çalıştığında seçim ekranında seçilebilir.

DATA gt\_spfli TYPE STANDARD TABLE OF spfli.  
PARAMETER : p\_varia TYPE disvariant-variant.  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_varia.  
PERFORM f4\_variant.  
START-OF-SELECTION.  
PERFORM alv\_goster.  
FORM alv\_goster .  
DATA : ls\_variant TYPE disvariant,  
lv\_repid   TYPE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
lv\_repid           = sy-repid.  
ls\_variant-report  = lv\_repid.  
ls\_variant-variant = p\_varia.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program = lv\_repid  
i\_structure\_name   = ‘SPFLI’  
i\_save             = ‘A’  
is\_variant         = ls\_variant  
TABLES  
t\_outtab           = gt\_spfli.  
ENDFORM.  
FORM f4\_variant .  
DATA: ls\_variant TYPE disvariant,  
lv\_save(1) TYPE c VALUE ‘A’,  
lv\_exit(1) TYPE c.  
CLEAR ls\_variant.  
ls\_variant-report = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_VARIANT\_F4′  
EXPORTING  
is\_variant    = ls\_variant  
i\_save        = lv\_save  
IMPORTING  
e\_exit        = lv\_exit  
es\_variant    = ls\_variant  
EXCEPTIONS  
not\_found     = 1  
program\_error = 2  
OTHERS        = 3.  
IF sy-subrc = 0 AND lv\_exit = space.  
p\_varia = ls\_variant-variant.  
ENDIF.  
ENDFORM.

P\_VARIA parametresi üzerinde F4 yardımı kullanıldığında, kaydedilmiş olan varaint listesi görünür. Bu listeden seçim yapılarak rapor çalıştığında düzen seçilen varaint’ a göre gelir.





**REUSE\_ALV\_VARIANT\_DEFAULT\_GET**

ALV raporuna ait varsayılan varaint’ ı elde etmek için kullanılır.

|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| I\_SAVE |  |  | SPACE | X | X | Varaint saklama seçeneği |
| **EXPORT** |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| CS\_VARIANT | LIKE | DISVARIANT |  |  | X | Variant bilgileri |
| **EXCEPTIONS** |  |  |  |  |  |  |
| **Exception** | **Tanım** |  |  |  |  |  |
| WRONG\_INPUT | Yanlış giriş paremetresi |  |  |  |  |  |
| NOT\_FOUND | Variant bulunamadı |  |  |  |  |  |
| PROGRAM\_ERROR | Program hatası |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Örnek: ALV’ ye ait varsayılan variant varsa bunu seçim ekranındaki variant parametresine taşıyan örnek.

DATA gt\_spfli TYPE STANDARD TABLE OF spfli.  
PARAMETER : p\_varia TYPE disvariant-variant.  
INITIALIZATION.  
PERFORM varsayilan\_varainti\_bul.  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_varia.  
PERFORM f4\_variant.  
START-OF-SELECTION.  
PERFORM alv\_goster.

FORM varsayilan\_varainti\_bul .  
DATA: ls\_variant TYPE disvariant,  
lv\_save(1) TYPE c VALUE ‘A’.  
CLEAR ls\_variant.  
ls\_variant-report = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_VARIANT\_DEFAULT\_GET’  
EXPORTING  
i\_save      = lv\_save  
CHANGING  
cs\_variant  = ls\_variant  
EXCEPTIONS  
wrong\_input         = 1  
not\_found           = 2  
program\_error       = 3  
OTHERS              = 4.  
IF sy-subrc = 0.  
p\_varia = ls\_variant-variant.  
ENDIF.  
ENDFORM.  
FORM f4\_variant .  
DATA: ls\_variant TYPE disvariant,  
lv\_save(1) TYPE c VALUE ‘A’,  
lv\_exit(1) TYPE c.  
CLEAR ls\_variant.  
ls\_variant-report = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_VARIANT\_F4′  
EXPORTING  
is\_variant    = ls\_variant  
i\_save        = lv\_save  
IMPORTING  
e\_exit        = lv\_exit  
es\_variant    = ls\_variant  
EXCEPTIONS  
not\_found     = 1  
program\_error = 2  
OTHERS        = 3.  
IF sy-subrc = 0 AND lv\_exit = space.  
p\_varia = ls\_variant-variant.  
ENDIF.  
ENDFORM.  
FORM alv\_goster .  
DATA : ls\_variant TYPE disvariant,  
lv\_repid   TYPE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
lv\_repid           = sy-repid.  
ls\_variant-report  = lv\_repid.  
ls\_variant-variant = p\_varia.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program = lv\_repid  
i\_structure\_name   = ‘SPFLI’  
i\_save             = ‘A’  
is\_variant         = ls\_variant  
TABLES  
t\_outtab           = gt\_spfli.  
ENDFORM.

Bu ALV’ ye ait varsayılan varaint “INITIALIZATION” sürecinde bulunarak ekrandaki “P\_VARIA” parametresine taşınır.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755a8de9033.png

**REUSE\_ALV\_VARIANT\_EXISTENCE**

Bir varaint’ ın mevcut olup olmadığı kontrolünü yapar.

|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| I\_SAVE |  |  | SPACE | X | X | Varaint saklama seçeneği |
| **CHANGING** |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| CS\_VARIANT | LIKE | DISVARIANT |  |  | X | Variant bilgileri |
| **EXCEPTIONS** |  |  |  |  |  |  |
| **Exception** | **Tanım** |  |  |  |  |  |
| WRONG\_INPUT | Yanlış giriş paremetresi |  |  |  |  |  |
| NOT\_FOUND | Variant bulunamadı |  |  |  |  |  |
| PROGRAM\_ERROR | Program hatası |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Örnek: Seçilen variant’ ın sistemde bulunup, bulunmadığını kontrol eden örnek.

DATA gt\_spfli TYPE STANDARD TABLE OF spfli.  
PARAMETER : p\_varia TYPE disvariant-variant.  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_varia.  
PERFORM f4\_variant.  
START-OF-SELECTION.  
PERFORM variant\_kontrolu.  
PERFORM alv\_goster.  
FORM f4\_variant .  
DATA: ls\_variant TYPE disvariant,  
lv\_save(1) TYPE c VALUE ‘A’,  
lv\_exit(1) TYPE c.  
CLEAR ls\_variant.  
ls\_variant-report = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_VARIANT\_F4′  
EXPORTING  
is\_variant    = ls\_variant  
i\_save        = lv\_save  
IMPORTING  
e\_exit        = lv\_exit  
es\_variant    = ls\_variant  
EXCEPTIONS  
not\_found     = 1  
program\_error = 2  
OTHERS        = 3.  
IF sy-subrc = 0 AND lv\_exit = space.  
p\_varia = ls\_variant-variant.  
ENDIF.  
ENDFORM.  
FORM variant\_kontrolu .  
DATA: ls\_variant TYPE disvariant,  
lv\_save(1) TYPE c VALUE ‘A’.  
IF p\_varia IS NOT INITIAL.  
ls\_variant-report   = sy-repid.  
ls\_variant-variant  = p\_varia.  
CALL FUNCTION ‘REUSE\_ALV\_VARIANT\_EXISTENCE’  
EXPORTING  
i\_save              = lv\_save  
CHANGING  
cs\_variant         = ls\_variant  
EXCEPTIONS  
wrong\_input         = 1  
not\_found           = 2  
program\_error       = 3  
OTHERS              = 4.  
IF sy-subrc <> 0.  
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
ENDIF.  
ENDIF.  
ENDFORM.                    ” VARIANT\_KONTROLU  
FORM alv\_goster .  
DATA : ls\_variant TYPE disvariant,  
lv\_repid   TYPE sy-repid.  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
lv\_repid           = sy-repid.  
ls\_variant-report  = lv\_repid.  
ls\_variant-variant = p\_varia.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program = lv\_repid  
i\_structure\_name   = ‘SPFLI’  
i\_save             = ‘A’  
is\_variant         = ls\_variant  
TABLES  
t\_outtab           = gt\_spfli.  
ENDFORM.

Seçim ekranında “P\_VARIA” parametresine yazılan varaint sistemde bulunuyor ise ALV bulunan variant düzeninde gösterilir. Bulunamaz ise hata mesajı verilir.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755ad12496d.png

**REUSE\_ALV\_EVENTS\_GET**

Lliste tipine göre ALV fonksiyonunun alabileceği bütün (CALLBACK) olayların listesini döner. Bu listedeki olaylara atanan alt yordam isimleri ALV fonksiyonuna verilerek alt yordamlar atınmış olur.

|  |
| --- |
| **IMPORT** |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Başlanıç d.** | **Ops.** | **Dğr atn.** | **Parametre açıklaması** |
| I\_LIST\_TYPE | TYPE | SLIS\_LIST\_TYPE | 0 | X | X | 1-4 arası liste tipi |
| **EXPORT** |  |  |  |  |  |  |
| **Parametre** | **Tanım tipi** | **İlişkili tip** | **Dğr atn.** | **Parametre açıklaması** |  |  |
| ET\_EVENTS | TYPE | SLIS\_T\_EVENT | X | Liste tipine göre olayların listesi |  |  |
| **EXCEPTIONS** |  |  |  |  |  |  |
| **Exception** | **Tanım** |  |  |  |  |  |
| LIST\_TYPE\_WRONG | Liste tipi mevcut değil |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

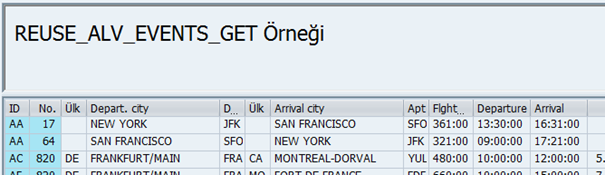
I\_LIST\_TYPE parametresinin alabileceği değerler ve açıklamaları aşağıdaki tablodadır.

|  |  |  |
| --- | --- | --- |
| **Parametre Değeri** | **ALV Liste Tipi** | **ALV Fonksiyonu** |
| 0 | Basit liste | REUSE\_ALV\_LIST\_DISPLAY |
| 1 | Hiyerarşik sıralı liste | REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY |
| 2 | Basit blok liste | REUSE\_ALV\_BLOCK\_LIST\_APPEND |
| 3 | Hiyerarşik sıralı blok liste | REUSE\_ALV\_BLOCK\_LIST\_HS\_APPEND |
| 4 | Tam ekran | REUSE\_ALV\_GRID\_DISPLAY |

Örnek: Aşağıdaki örnekte hücresel ALV gösterimi için ALV fonksiyonunda “I\_CALLBACK\_TOP\_OF\_PAGE” parametresi kullanılmadı. Bunun yerine “REUSE\_ALV\_EVENTS\_GET” fonksiyonu kullanılarak “IT\_EVENTS” tablosu alınır. Bu tabloda olaya alt yordam ismi atanır, ALV fonksiyonuna bu tablo parametre olarak verilir.

TYPE-POOLS: slis.  
DATA: gt\_spfli TYPE STANDARD TABLE OF spfli,  
gv\_repid LIKE sy-repid,  
gt\_events TYPE slis\_t\_event.  
START-OF-SELECTION.  
PERFORM alv\_callback\_olaylari\_bul.  
PERFORM alv\_goster.  
FORM alv\_callback\_olaylari\_bul .  
DATA ls\_events TYPE slis\_alv\_event.  
CALL FUNCTION ‘REUSE\_ALV\_EVENTS\_GET’  
EXPORTING  
i\_list\_type           = 4  
IMPORTING  
et\_events             = gt\_events  
EXCEPTIONS  
list\_type\_wrong       = 1  
OTHERS                = 2.

IF sy-subrc <> 0.  
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
ENDIF.  
READ TABLE gt\_events INTO ls\_events  
WITH KEY name = slis\_ev\_top\_of\_page.  
IF sy-subrc = 0.  
ls\_events-form = ‘TOP\_OF\_PAGE’.  
MODIFY gt\_events FROM ls\_events INDEX sy-tabix.  
ENDIF.  
ENDFORM.  
FORM alv\_goster .  
SELECT \* FROM spfli INTO CORRESPONDING FIELDS OF TABLE gt\_spfli.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
i\_structure\_name    = ‘SPFLI’  
i\_save              = ‘A’  
it\_events           = gt\_events  
TABLES  
t\_outtab            = gt\_spfli.  
ENDFORM.  
FORM top\_of\_page.  
DATA: lt\_header    TYPE slis\_t\_listheader,  
ls\_header    TYPE slis\_listheader.  
ls\_header-typ  = ‘H’.  
ls\_header-key  = ”.  
ls\_header-info = ‘REUSE\_ALV\_EVENTS\_GET Örneği’.  
APPEND ls\_header TO lt\_header.  
CALL FUNCTION ‘REUSE\_ALV\_COMMENTARY\_WRITE’  
EXPORTING  
it\_list\_commentary  = lt\_header  
i\_logo              = ‘ENJOYSAP\_LOGO’.  
ENDFORM.



**Sütun Görüntüsü Biçimlendirme**

Alan katalogu özellikleri değiştirilerek sütuna bağlı alanların görüntüsü değiştirilebilir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Alan** | **Anlamı** | **Veri Tipi** | **Değer Aralığı** |
| HOTSPOT | Alan üzerinde geldiğinde imlecin el şeklini (link) almasını sağlar | CHAR(1) | SPACE, ‘X’ |
| CHECKBOX | Alanın checkbox olarak görünmesini sağlar | CHAR(1) | SPACE, ‘X’ |
| DO\_SUM | ALV’ de belirtilen alanın alt toplamı alınarak görüntülenir | CHAR(1) | SPACE, ‘X’ |
| COL\_POS | Alanın sütun pozisyonu ayarlamak için kullanılır | INT4(10) | Tamsayı |
| EMPHASIZE | Sütun rengini değiştirmek için kullanılır‘X’ kullanılır ise ön tanımlı tipler kullanılır. Eğer karakter ‘C’(renk kodu) ile başlarsa, alabileceği üç rakamın anlamı şöyledir.1. x: renk kodu2. y: yoğunluk görünüm3. z: ters çevrilmiş renk görünümü | CHAR(4) | SPACE, ‘X’ or ‘Cxyz’ (x:’1′-‘9′; y,z: ‘0’=off ‘1’=on) |
| NO\_OUT | Alan gizlenir. Alan alan listesinde kalmaya devam eder. ALV görüntüledikten sonra düzen içerisinden listeye dâhil edilebilir | CHAR(1) | SPACE, ‘X’ |
| OUTPUTLEN | Alanın görüntülenen sütun genişliğini ayarlamak için kullanılır. ABAP Dictionary’ de tanımlı bir alan ise doldurmaya gerek yoktur. Alan uzunluğu Domain üzerinden devralınır. | NUMC(6) | Tamsayı |

**Sütun İçeriğini Biçimlendirme**

Alan katalogu özellikleri değiştirilerek sütuna bağlı alanların içerikleri biçimlendirilebilir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Alan** | **Anlamı** | **Veri Tipi** | **Değer Aralığı** |
| EDIT\_MASK | Geçerli alan çıktısı için çeviri yordamı atanmış ise kullanılmasını sağlar. | CHAR(60) | SPACE, çeviri yordamı |
| ICON | Sütun içeriği ikon olarak görüntülenir. Sütun içeriği geçerli ikon karakter dizesi içermelidir. | CHAR(1) | SPACE, ‘X’ |
| JUST | Hizalama için kullanılır. CHAR ve NUMC veri tipine sahip alanlara uygulanabilir.· ‘R’: Sağa dayalı· ‘L’: Sola dayalı· ‘R’: Ortalanmış | CHAR(1) | SPACE, ‘R’, ‘L’,’C’ |
| LZERO | NUMC veri tipi için kullanılabilir. Alanın sıfırdan büyük rakamının solunda sıfır var ise görüntülenir. | CHAR(1) | SPACE, ‘X’ |
| NO\_SIGN | Alan değeri işaretsiz olarak görüntülenir. | CHAR(1) | SPACE, ‘X’ |
| NO\_ZERO | Sıfır alanlar boş olarak görüntülenir. | CHAR(1) | SPACE, ‘X’ |
| SYMBOL | Sütun içeriği sembol olarak görüntülenir. Alan içeriği geçerli sembol işaretini içermelidir. | CHAR(1) | SPACE, ‘X’ |

**Sütun Metin Özelliklerini Biçimlendirme**

Alan katalogu özellikleri değiştirilerek sütun başlığı, ipucu yardımı, sütun seçim metinlerinin bakımı yapılabilir.

LVC\_S\_FCAT için metin biçimlendirme özellikleri

|  |  |  |  |
| --- | --- | --- | --- |
| **Alan** | **Anlamı** | **Veri Tipi** | **Değer Aralığı** |
| TOOLTIP | Sütuna bağlı ipucu yardımı metni | CHAR(40) | Herhangi bir metin |
| TIPDDICTXT | ABAP Data Dictionary’ e referans gösterilen alanlar için metin seçimi. | CHAR(1) | SPACE, ‘L’, ‘M’, ‘S’ ve ‘R’ |
| SELTEXT | Sütun seçiminde kullanılacak metin | CHAR(40) | Herhangi bir metin |
| SELDDICTXT | ABAP Dictionary’ e referans gösterilen alanlar için sütun seçiminde kullanılacak metin | CHAR(1) | SPACE, ‘L’, ‘M’, ‘S’ ve ‘R’ |
| SCRTEXT\_S | ABAP Dictionary’ de tanımlı alanlar için tanımlı kısa metin bu alana kopyalanır. | CHAR(10) | ABAP Dictionary’ den kopyalanan metin |
| SCRTEXT\_M | ABAP Dictionary’ de tanımlı alanlar için tanımlı orta uzunluktaki metin bu alana kopyalanır. | CHAR(20) | ABAP Dictionary’ den kopyalanan metin |
| SCRTEXT\_L | ABAP Dictionary’ de tanımlı alanlar için tanımlı uzun metin bu alana kopyalanır. | CHAR(40) | ABAP Dictionary’ den kopyalanan metin |
| REPTEXT | ABAP Dictionary’ de tanımlı alanlar için uygun data element’ alan etiketi bu alana kopyalanır. | CHAR(55) | ABAP Dictionary’ den kopyalanan metin |
| COLTEXT | ABAP Dictionary’ e referans gösterilmeyen alan için sütun başlığı belirler. | CHAR(40) | Herhangi bir metin |
| COLDDICTXT | ABAP Dictionary’ e referans gösterilen alanlar için sütun başlığı seçiminde kullanılacak metin. | CHAR(1) | SPACE, ‘L’, ‘M’, ‘S’ ve ‘R’ |

SLIS\_FIELDCAT\_ALV için metin biçimlendirme özellikleri

|  |  |  |  |
| --- | --- | --- | --- |
| **Alan** | **Anlamı** | **Veri Tipi** | **Değer Aralığı** |
| SELTEXT\_S | ABAP Dictionary’ de tanımlı alanlar için tanımlı kısa metin bu alana kopyalanır. | CHAR(10) | ABAP Dictionary’ den kopyalanan metin |
| SELTEXT\_M | ABAP Dictionary’ de tanımlı alanlar için tanımlı orta uzunluktaki metin bu alana kopyalanır. | CHAR(20) | ABAP Dictionary’ den kopyalanan metin |
| SELTEXT\_L | ABAP Dictionary’ de tanımlı alanlar için tanımlı uzun metin bu alana kopyalanır. | CHAR(40) | ABAP Dictionary’ den kopyalanan metin |
| REPTEXT | ABAP Dictionary’ de tanımlı alanlar için uygun data element’ alan etiketi bu alana kopyalanır. | CHAR(55) | ABAP Dictionary’ den kopyalanan metin |

Örnek: Sütun biçimlendirme uygulanan örnek.

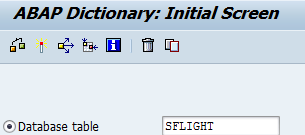
TYPE-POOLS: slis.  
DATA: gt\_fieldcat  TYPE slis\_t\_fieldcat\_alv,  
gs\_fieldcat  TYPE slis\_fieldcat\_alv.  
TYPES: BEGIN OF gy\_sbook,  
carrid TYPE sbook-carrid,  
connid TYPE sbook-connid,  
fldate TYPE sbook-fldate,  
bookid TYPE sbook-bookid,  
passname TYPE sbook-passname,  
icon   TYPE c LENGTH 4,  
END OF gy\_sbook.  
DATA: gt\_sbook TYPE STANDARD TABLE OF gy\_sbook,  
gs\_sbook TYPE gy\_sbook.  
DATA gv\_repid TYPE sy-repid.  
SELECT \* FROM sbook INTO CORRESPONDING FIELDS OF TABLE gt\_sbook.  
LOOP AT gt\_sbook INTO gs\_sbook.  
gs\_sbook-icon = ‘@CQ@’.  
MODIFY gt\_sbook FROM gs\_sbook.  
ENDLOOP.  
CLEAR gs\_fieldcat.  
gs\_fieldcat-fieldname  = ‘CARRID’.  
gs\_fieldcat-seltext\_m  = ‘Hava Alanı’.  
gs\_fieldcat-hotspot    = ‘X’.  
gs\_fieldcat-just       = ‘R’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
CLEAR gs\_fieldcat.  
gs\_fieldcat-fieldname  = ‘CONNID’.  
gs\_fieldcat-seltext\_m  = ‘Bağlantı No.’.  
gs\_fieldcat-emphasize  = ‘C601′.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
CLEAR gs\_fieldcat.  
gs\_fieldcat-fieldname  = ‘FLDATE’.  
gs\_fieldcat-seltext\_m  = ‘Tarih’.  
gs\_fieldcat-emphasize  = ‘X’.  
gs\_fieldcat-COL\_POS    = 5.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
CLEAR gs\_fieldcat.  
gs\_fieldcat-fieldname  = ‘BOOKID’.  
gs\_fieldcat-seltext\_m  = ‘Rezervasyon ID’.  
gs\_fieldcat-outputlen  = 15.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
CLEAR gs\_fieldcat.  
gs\_fieldcat-fieldname  = ‘PASSNAME’.  
gs\_fieldcat-seltext\_m  = ‘Yolcu İsmi’.  
gs\_fieldcat-no\_out     = ‘X’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
CLEAR gs\_fieldcat.  
gs\_fieldcat-fieldname  = ‘ICON’.  
gs\_fieldcat-seltext\_m  = ‘Icon’.  
gs\_fieldcat-icon       = ‘X’.  
APPEND gs\_fieldcat TO gt\_fieldcat.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program  = gv\_repid  
it\_fieldcat         = gt\_fieldcat  
i\_save              = ‘A’  
TABLES  
t\_outtab            = gt\_sbook  
EXCEPTIONS  
program\_error       = 1  
OTHERS              = 2.



Örnek: Hotspot

TYPE-POOLS: slis.  
DATA : gt\_mara TYPE STANDARD TABLE OF mara,  
gs\_mara TYPE mara,  
gv\_repid TYPE sy-repid.  
DATA : gt\_fieldcat TYPE slis\_t\_fieldcat\_alv,  
gs\_fieldcat TYPE slis\_fieldcat\_alv.  
SELECT \* FROM mara UP TO 10 ROWS  
INTO CORRESPONDING FIELDS OF TABLE gt\_mara.  
gv\_repid = sy-repid.  
CALL FUNCTION ‘REUSE\_ALV\_FIELDCATALOG\_MERGE’  
EXPORTING  
i\_program\_name           = gv\_repid  
i\_structure\_name         = ‘MARA’  
CHANGING  
ct\_fieldcat             = gt\_fieldcat  
EXCEPTIONS  
inconsistent\_interface   = 1  
program\_error            = 2  
OTHERS                   = 3.  
gs\_fieldcat-hotspot = ‘X’.  
MODIFY gt\_fieldcat FROM gs\_fieldcat  
TRANSPORTING hotspot  
WHERE fieldname = ‘MATNR’.  
CALL FUNCTION ‘REUSE\_ALV\_GRID\_DISPLAY’  
EXPORTING  
i\_callback\_program      = gv\_repid  
it\_fieldcat             = gt\_fieldcat  
i\_save                  = ‘A’  
i\_callback\_user\_command = ‘USER\_COMMAND’  
TABLES  
t\_outtab                = gt\_mara  
EXCEPTIONS  
program\_error           = 1  
OTHERS                  = 2.  
FORM user\_command  USING r\_ucomm LIKE sy-ucomm  
rs\_selfield TYPE slis\_selfield.  
CASE r\_ucomm.  
WHEN ‘&IC1′.  
READ TABLE gt\_mara INTO gs\_mara INDEX rs\_selfield-tabindex.  
IF sy-subrc EQ 0.  
SET PARAMETER ID ‘MAT’ FIELD gs\_mara-matnr.  
CALL TRANSACTION ‘MM03′ AND SKIP FIRST SCREEN.  
ENDIF.  
ENDCASE.  
ENDFORM.                    ” USER\_COMMAND

Örnek:

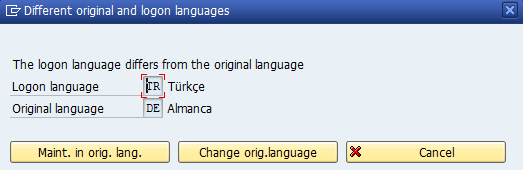


Copy…http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755c5bd012e.png (Ctrl+F5)



Package ve request işlemleri yapılır.

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755c7348678.png



http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755c8b7d4de.png

http://iuyanik.com/abaptr/wp-content/uploads/2013/11/img_52755c8c08980.png

