**BATCH INPUT**

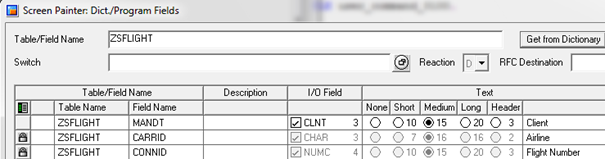
Toplu veri girişi yapmak için kullanılır. İşlem kodunu çalıştırılır ve kullanıcı verileri giriyormuş gibi ekrandaki alanlara veriler transfer edilir. Alana girilen metin, imleç pozisyonu, tıklanılan buton gibi bilgiler kaydedilir. Örnek giriş kaydı SHDB işlem kodu kaydedilebilir. SM35 işlem kodu ile toplu girdi oturumları izlenilir.

Örnek: ZSFLIGHT tablosuna batch input ile kayıt ekleyen program örneği. Örnek üç aşamalıdır.

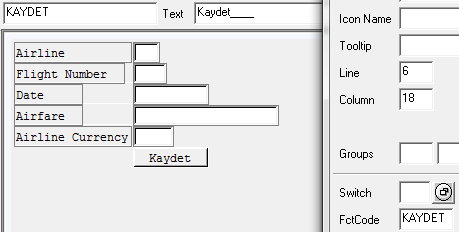
1. ZSFLIGHT tablosuna kayıt ekleyen programın hazırlanması (Standart işlem kodlarına veri aktarımı yapılacağı zaman bu adıma gerek yoktur).
2. SHDB ile örnek toplu girdi örneği hazırlanması,
3. Toplu girdi işlemini yapacak program hazırlanması

ZSFLIGHT tablosuna kayıt ekleyen programın hazırlanması.

Ekran 100’ ü açın. F6 tuşuna basarak alan seçim ekranını açın. Table/Field Name alanına ZSFLIGHT tablosunu yazın ve Get from Dictionary butonuna tıklayın. ZSFLIGHT tablosuna bağlı alanlar listelenecektir.



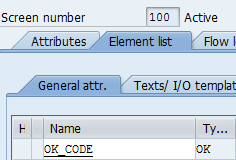
Alan seçim penceresinden ilke beş alanı seçin ve ekrana sürükleyin.

Ekrana bir tane buton ekleyin. FctCode alanına KAYDET girin. 

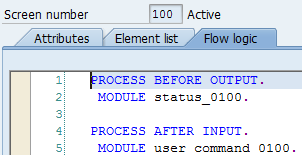
GUI Status nesnesi oluşturun ve alanlarını aşağıdaki şekilde kaydedin.

C:\Users\grencuz\Desktop\untitled.png

100 ekranı Element list sekmesinden OK kodu alanına OK\_CODE ismini girin.



Flow logic sekmesinde yorum satırlarını kaldırın.



Aşağıdaki kodu ekleyin.

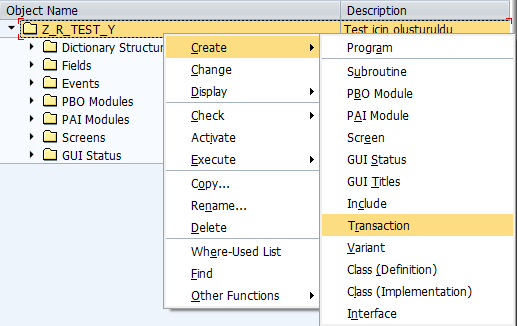
TABLES zsflight.  
DATA: ok\_code LIKE sy-ucomm.

START-OF-SELECTION.  
CALL SCREEN 100.

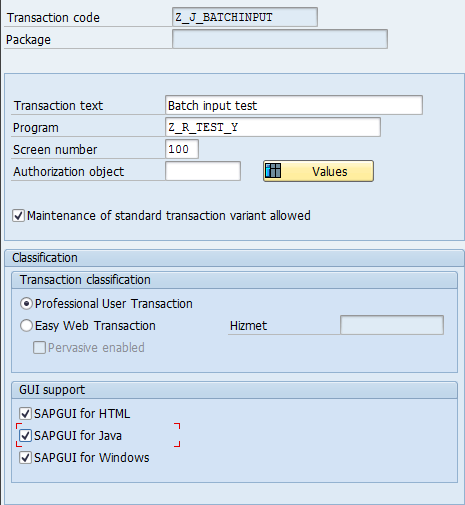
MODULE status\_0100 OUTPUT.  
SET PF-STATUS ‘100’.  
ENDMODULE.

MODULE user\_command\_0100 INPUT.  
CASE ok\_code.  
WHEN ‘GERI’ OR ‘CIKIS’ OR ‘IPTAL’.  
LEAVE TO SCREEN 0.  
WHEN ‘&KAYDET’.  
MODIFY zsflight FROM zsflight.  
COMMIT WORK AND WAIT.  
MESSAGE ‘Kayıtlar saklandı.’ TYPE ‘S’.  
ENDCASE.  
ENDMODULE.

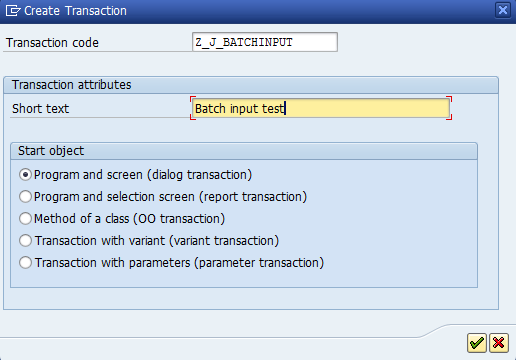
Programa işlem kodu ekleyin. Nesne listesi içerisinde program ismine sağ tıklayın ve Create->Transaction seçeneğini seçin.



Create Transaction penceresi açılacaktir. Alanları aşağıdaki şekilde doldurun ve devam edin.



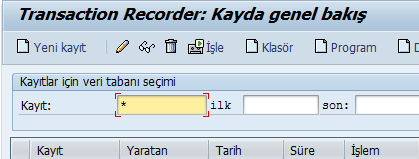
İşlem kodu özellikleri açılacaktır. Alanları resimdeki gibi doldurun ve kaydedin.



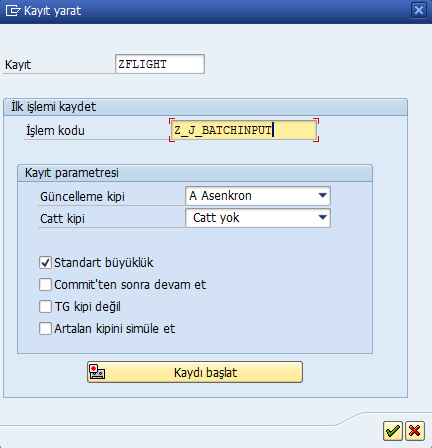
Package’ a bağlama ve request alma işlemleri sorulacaktır. Çalışma şeklinize göre ilerleyin.

SHDB ile örnek toplu girdi örneği oluşturulması

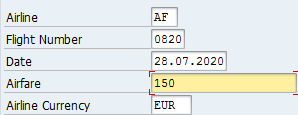
SHDB işlem kodunu çalıştırın. Yeni Kayıt butonuna tıklayın.



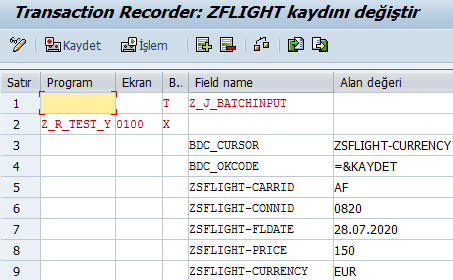
Kayıt yarat ekranı açılacaktır. Alanları resimdeki gibi doldurun. Kaydı başlat butonuna tıklayın.



Kayıt yapılacak işlem kodu çalıştırılır. İlk adımda tanımlanan 100 ekranı açılır. Ekran alanlarını doldurun ve kaydet butonuna  tıklayın.



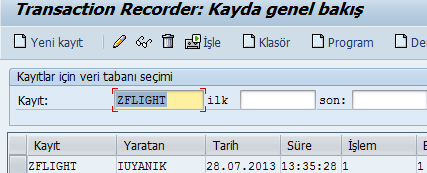
Toplu girdi kaydına ait bilgiler Transaction Recorder ekranına gelir.



Bu ekranda değişiklik yapmadan kaydet C:\Users\bck\AppData\Local\Microsoft\Windows\INetCache\Content.Word\indir.png butonuna tıklayın. Toplu girdi kaydı oluşturulacaktır.



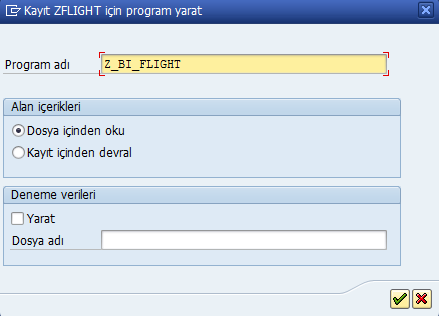
Toplu giriş kaydını daha sonra üzerine çift tıklayarak içeriğini görüntüleyebilirsiniz.



Oluşturulan kayıttan örnek program oluşturulabilir. Bunun kayıt seçilir ve Program butonuna tıklanır.

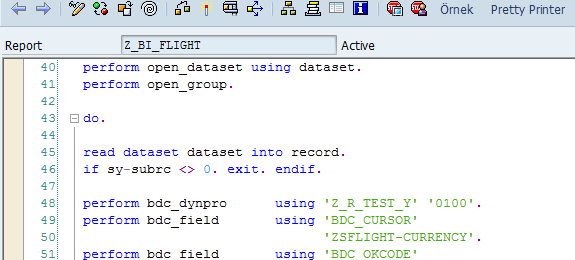


Açılan ekranda yeni programın adı girilir ve devam edilir.



Program özellikleri ekranı açılacaktır. Program tipini 1 (Yürütülebilir program) olarak seçip, Source code butonuna tıklayın.

Package’ a bağlama ve request alma işlemlerini yapın. Bu adımdan sonra program kodu açılır. Program oluşturarak alanları kendi programınıza kolay bir şekilde aktarabilirsiniz.



Toplu girdi işlemini yapacak program

DATA : g\_docking\_container TYPE REF TO cl\_gui\_docking\_container,  
g\_alv\_grid          TYPE REF TO cl\_gui\_alv\_grid,  
ok\_code LIKE sy-ucomm,  
gt\_sflight TYPE STANDARD TABLE OF sflight,  
gs\_sflight TYPE sflight.

DATA: gt\_bdcdata TYPE STANDARD TABLE OF bdcdata,  
gs\_bdcdata TYPE bdcdata.

CLASS lcl\_alv DEFINITION.  
PUBLIC SECTION.  
METHODS:  
alv\_olustur,

handle\_toolbar  
FOR EVENT toolbar OF cl\_gui\_alv\_grid  
IMPORTING e\_object  
e\_interactive  
sender,

handle\_user\_command  
FOR EVENT user\_command OF cl\_gui\_alv\_grid  
IMPORTING e\_ucomm sender.  
ENDCLASS.

DATA g\_alv TYPE REF TO lcl\_alv.

CLASS lcl\_alv IMPLEMENTATION.  
METHOD alv\_olustur.  
DATA: lt\_fieldcat   TYPE lvc\_t\_fcat,  
ls\_fieldcat   TYPE lvc\_s\_fcat,  
ls\_variant    TYPE disvariant,  
ls\_layout     TYPE lvc\_s\_layo,  
lt\_toolbar\_excluding TYPE ui\_functions,  
lv\_repid      TYPE sy-repid.

DATA: lt\_cells TYPE lvc\_t\_cell,  
ls\_cells TYPE lvc\_s\_cell.

IF g\_docking\_container IS INITIAL.

CREATE OBJECT g\_docking\_container  
EXPORTING  
side       = g\_docking\_container->dock\_at\_top  
extension  = 2000.

CREATE OBJECT g\_alv\_grid  
EXPORTING  
i\_parent          = g\_docking\_container  
EXCEPTIONS  
error\_cntl\_create = 1  
error\_cntl\_init   = 2  
error\_cntl\_link   = 3  
error\_dp\_create   = 4  
OTHERS            = 5.  
IF sy-subrc <> 0.  
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
ENDIF.

CALL FUNCTION ‘LVC\_FIELDCATALOG\_MERGE’  
EXPORTING  
i\_structure\_name        = ‘ZSFLIGHT’  
CHANGING  
ct\_fieldcat             = lt\_fieldcat  
EXCEPTIONS  
inconsistent\_interface  = 1  
program\_error           = 2  
OTHERS                  = 3.

IF sy-subrc <> 0.  
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
ENDIF.

LOOP AT lt\_fieldcat INTO ls\_fieldcat.  
ls\_fieldcat-edit = ‘X’.  
MODIFY lt\_fieldcat FROM ls\_fieldcat.  
ENDLOOP.

ls\_layout-box\_fname    = ‘BOX’.

lv\_repid = sy-repid.  
ls\_variant-report      = lv\_repid.

CALL METHOD g\_alv\_grid->set\_table\_for\_first\_display  
EXPORTING  
i\_save                        = ‘A’  
is\_variant                    = ls\_variant  
is\_layout                     = ls\_layout  
it\_toolbar\_excluding          = lt\_toolbar\_excluding  
CHANGING  
it\_outtab                     = gt\_sflight  
it\_fieldcatalog               = lt\_fieldcat  
EXCEPTIONS  
invalid\_parameter\_combination = 1  
program\_error                 = 2  
too\_many\_lines                = 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.

SET HANDLER me->handle\_toolbar FOR g\_alv\_grid.  
SET HANDLER me->handle\_user\_command FOR g\_alv\_grid.

CALL METHOD cl\_gui\_control=>set\_focus  
EXPORTING  
control = g\_alv\_grid.

CALL METHOD g\_alv\_grid->set\_ready\_for\_input  
EXPORTING  
i\_ready\_for\_input = 1.

ELSE.  
CALL METHOD g\_alv\_grid->refresh\_table\_display.  
CALL METHOD cl\_gui\_cfw=>flush.  
ENDIF.

ENDMETHOD.

METHOD handle\_toolbar.  
DATA ls\_toolbar TYPE stb\_button.

CLEAR ls\_toolbar. MOVE 3 TO ls\_toolbar-butn\_type.  
APPEND ls\_toolbar TO e\_object->mt\_toolbar.

CLEAR ls\_toolbar.  
MOVE ‘BATCHINPUT’       TO ls\_toolbar-function.  
MOVE icon\_activity      TO ls\_toolbar-icon.  
MOVE ‘Batch input’      TO ls\_toolbar-text.  
MOVE ‘Batch input’      TO ls\_toolbar-quickinfo.  
APPEND ls\_toolbar       TO e\_object->mt\_toolbar.  
ENDMETHOD.

METHOD handle\_user\_command.

DATA: et\_index\_rows TYPE lvc\_t\_row,  
et\_row\_no     TYPE lvc\_t\_roid,  
wa\_row\_no     LIKE LINE OF et\_row\_no.

DATA: ls\_sflight TYPE sflight,  
lv\_price   type c length 18.

DATA: lt\_messtab TYPE STANDARD TABLE OF bdcmsgcoll,  
ls\_messtab TYPE bdcmsgcoll.

CASE e\_ucomm.  
WHEN ‘BATCHINPUT’.

CALL METHOD g\_alv\_grid->get\_selected\_rows  
IMPORTING  
et\_index\_rows = et\_index\_rows  
et\_row\_no     = et\_row\_no.

LOOP AT et\_row\_no INTO wa\_row\_no  .  
READ TABLE gt\_sflight INTO ls\_sflight INDEX wa\_row\_no-row\_id.  
PERFORM bdc\_dynpro      USING ‘Z\_R\_TEST\_Y’        ‘0100’.  
PERFORM bdc\_field       USING ‘BDC\_CURSOR’        ‘ZSFLIGHT-CURRENCY’.  
PERFORM bdc\_field       USING ‘BDC\_OKCODE’        ‘=&KAYDET’.  
PERFORM bdc\_field       USING ‘ZSFLIGHT-CARRID’   ls\_sflight-carrid.  
PERFORM bdc\_field       USING ‘ZSFLIGHT-CONNID’   ls\_sflight-connid.  
PERFORM bdc\_field       USING ‘ZSFLIGHT-FLDATE’   ls\_sflight-fldate.

WRITE ls\_sflight-price TO lv\_price.  
PERFORM bdc\_field       USING ‘ZSFLIGHT-PRICE’    lv\_price.  
PERFORM bdc\_field       USING ‘ZSFLIGHT-CURRENCY’ ls\_sflight-currency.  
CALL TRANSACTION ‘Z\_J\_BATCHINPUT’ USING gt\_bdcdata  
MODE ‘A’  
UPDATE ‘S’  
MESSAGES INTO lt\_messtab.  
ENDLOOP.

LOOP AT lt\_messtab INTO ls\_messtab WHERE msgtyp = ‘AEX’.  
EXIT.  
ENDLOOP.

IF sy-subrc = 0.  
MESSAGE ID ls\_messtab-msgid TYPE ls\_messtab-msgtyp NUMBER ls\_messtab-msgnr  
WITH ls\_messtab-msgv1 ls\_messtab-msgv2 ls\_messtab-msgv3 ls\_messtab-msgv4.  
ELSE.  
READ TABLE lt\_messtab INTO ls\_messtab INDEX 1.  
IF sy-subrc = 0.  
MESSAGE ID ls\_messtab-msgid TYPE ls\_messtab-msgtyp NUMBER ls\_messtab-msgnr  
WITH ls\_messtab-msgv1 ls\_messtab-msgv2 ls\_messtab-msgv3 ls\_messtab-msgv4.  
ENDIF.  
ENDIF.

CALL METHOD sender->refresh\_table\_display.  
ENDCASE.

ENDMETHOD.  
ENDCLASS.

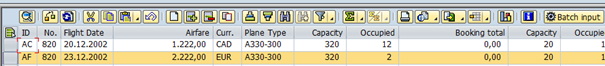
START-OF-SELECTION.  
SELECT \* FROM sflight  
INTO CORRESPONDING FIELDS OF TABLE gt\_sflight UP TO 10 ROWS.  
CREATE OBJECT g\_alv.  
CALL SCREEN 0100.

MODULE status\_0100 OUTPUT.  
SET PF-STATUS ‘100’.  
CALL METHOD g\_alv->alv\_olustur.  
ENDMODULE.

MODULE user\_command\_0100 INPUT.  
CASE ok\_code.  
WHEN ‘GERI’ OR ‘CIKIS’ OR ‘IPTAL’.  
LEAVE TO SCREEN 0.  
ENDCASE.  
ENDMODULE.

FORM bdc\_dynpro USING program dynpro.  
CLEAR gs\_bdcdata.  
gs\_bdcdata-program  = program.  
gs\_bdcdata-dynpro   = dynpro.  
gs\_bdcdata-dynbegin = ‘X’.  
APPEND gs\_bdcdata TO gt\_bdcdata.  
ENDFORM.

FORM bdc\_field USING fnam fval.  
CLEAR gs\_bdcdata.  
gs\_bdcdata-fnam = fnam.  
gs\_bdcdata-fval = fval.  
APPEND gs\_bdcdata TO gt\_bdcdata.  
ENDFORM.



ZSFLIGHT tablosuna yeni kayıt gelmiştir.

