**SALES CONTRACT CREATION USING BAPI**

**Program for creation of sales contract and displaying the created sales contract details and status in OOALV format.**

\*&---------------------------------------------------------------------\*  
\*& Report ZGRP5\_PROG1  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ZGRP5\_PROG1.  
  
\*----------------------------------------------------------------------\*  
\*    TOP INCLUDE FOR DECLARATIONS  
\*----------------------------------------------------------------------\*  
include zgrp5\_top.  
  
  
\*----------------------------------------------------------------------\*  
\*    INCLUDE FOR  SELECTION-SCREEN  
\*----------------------------------------------------------------------\*  
include zgrp5\_selectionscreen.  
  
  
\*----------------------------------------------------------------------\*  
\*     START-OF-SELECTION  
\*----------------------------------------------------------------------\*  
START-OF-SELECTION.  
  
  
\*----------------------------------------------------------------------\*  
\*    INCLUDE FOR UPLOADING THE CSV FILE AND PROCESSING  
\*----------------------------------------------------------------------\*  
include zgrop\_5\_display\_op.  
  
  
  
\*----------------------------------------------------------------------\*  
\*    INCLUDE FOR CREATION OF CONTRACT USING BAPI FUNCTION MODULES  
\*----------------------------------------------------------------------\*  
include zgrp5\_mainprog.  
  
  
\*----------------------------------------------------------------------\*  
\*    INCLUDE FOR DISPLAYING THE OOALV SCREEN  
\*----------------------------------------------------------------------\*  
  
INCLUDE ZGRP\_OOALVSCREEN.  
  
  
\*----------------------------------------------------------------------\*  
\*    INCLUDE FOR DISPLAYING THE CONTRACT DETAILS IN VA43  
\*----------------------------------------------------------------------\*  
INCLUDE ZGRP5\_GETCURSOR.  
  
  
\*----------------------------------------------------------------------\*  
\*     INCLUDE FOR MODULE DEFINATIONS FOR OOALV OUTPUT  
\*----------------------------------------------------------------------\*  
  
\*include zgrp4\_alv.  
  
include zgrp5\_alv.

**INCLUDE FOR UPLOADING THE CSV FILE AND PROCESS THE DATA FROM CSV**

\*&---------------------------------------------------------------------\*  
\*& Include          ZGRP5\_TOP  
\*&---------------------------------------------------------------------\*  
TYPE-POOLS: kcde.  
  
CONSTANTS: c\_separator TYPE c VALUE ','.  
  
DATA: gt\_intern    TYPE kcde\_intern,  
      gwa\_intern   TYPE kcde\_intern\_struc.  
  
DATA: gt\_input     TYPE TABLE OF ZG05\_STRUCTURE,  
      gwa\_input    TYPE ZG05\_STRUCTURE,  
      it\_tab22    type TABLE of ZG05\_STRUCTURE.  
  
  
DATA: ls\_cont\_hdr\_in TYPE bapisdhd1,  
             ls\_cont\_hdr\_inx TYPE bapisdhd1x,  
             lv\_salesdoc TYPE bapivbeln-vbeln,  
              lt\_return TYPE TABLE OF bapiret2,  
             lt\_partner TYPE TABLE OF bapiparnr,  
             lt\_item\_in TYPE TABLE OF bapisditm,  
             lt\_item\_inx TYPE TABLE OF bapisditmx,  
            lv\_msg TYPE string,  
            LV\_VAR1 TYPE STRING,  
            WA\_FIELDCAT TYPE lvc\_s\_fcat,  " ALV FCAT FOR WA  
            LT\_FIELDCAT TYPE lvc\_T\_Fcat.   
            data : i\_container  type ref to cl\_gui\_custom\_container,  
                   i\_grid       type ref to cl\_gui\_alv\_grid.  
  
FIELD-SYMBOLS <FS\_DATA> TYPE ANY.

**TYPE-POOLS:** kcde.: This statement declares the type pool "kcde", which means the program can now use data types defined in the "kcde" type pool.

**CONSTANTS:** c\_separator TYPE c VALUE ','.: This line defines a constant named "c\_separator" of type character (c) with the value ',' (comma). It's likely used as a separator in string operations.

**DATA:** This section declares various data objects such as internal tables, structures, and variables.

**gt\_intern, gwa\_intern:** These are data objects declared using the types from the "kcde" type pool. They seem to be related to internal structures or tables.

**gt\_input, gwa\_input, it\_tab22:** These are internal tables (TYPE TABLE OF) or structures (TYPE) defined either by referencing custom-defined structures (like ZG05\_STRUCTURE) or SAP standard structures.

**ls\_cont\_hdr\_in, ls\_cont\_hdr\_inx, lv\_salesdoc, lt\_return, lt\_partner, lt\_item\_in, lt\_item\_inx, lv\_msg, LV\_VAR1**: These are individual data objects declared using predefined SAP data types or custom types.

**WA\_FIELDCAT, LT\_FIELDCAT:** These seem to be related to ALV (ABAP List Viewer) functionality. They're used for field catalog operations in ALV output.

**FIELD-SYMBOLS <FS\_DATA> TYPE ANY:** This statement declares a field symbol <FS\_DATA> that can point to any data type (TYPE ANY). Field symbols are used for dynamic assignment of data objects at runtime.

**INCLUDE FOR F4 FUNCTIONALITY(SEARCH HELP):**

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZGRP5\_SELECTIONSCREEN*  
*\*&---------------------------------------------------------------------\**  
  
  
PARAMETERS: p\_file TYPE localfile OBLIGATORY.  
*\*----------------------------------------------------------------------\**  
*\*     AT SELECTION-SCREEN ON VALUE-REQUEST*  
*\*----------------------------------------------------------------------\**  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_file.  
*\*F4 help for the file in selection screen*  
  CALL FUNCTION 'F4\_FILENAME'  
    IMPORTING  
      file\_name = p\_file.

**INCLUDE FOR UPLOADING THE CSV FILE AND PROCESSING:**

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZGROP\_5\_DISPLAY\_OP*  
*\*&---------------------------------------------------------------------\**  
DATA: lv\_filename TYPE rlgrap-filename.  
  DATA: lv\_index TYPE i.  
  FIELD-SYMBOLS <gt\_data> TYPE ANY.  
  
  lv\_filename = p\_file.  
  
*\*Read the CSV file data to an internal table*  
  CALL FUNCTION 'KCD\_CSV\_FILE\_TO\_INTERN\_CONVERT'  
    EXPORTING  
      i\_filename      = lv\_filename  
      i\_separator     = c\_separator  
    TABLES  
      e\_intern        = gt\_intern  
    EXCEPTIONS  
      upload\_csv      = 1  
      upload\_filetype = 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 gt\_intern INTO gwa\_intern.  
    MOVE : gwa\_intern-col TO lv\_index.  
    ASSIGN COMPONENT lv\_index OF STRUCTURE gwa\_input TO <gt\_data>.  
    MOVE : gwa\_intern-value TO <gt\_data>.  
    AT END OF row.  
      APPEND gwa\_input TO gt\_input.  
      CLEAR gwa\_input.  
    ENDAT.  
  ENDLOOP.

**INCLUDE FOR CREATION OF CONTRACT USING BAPI FUNCTION MODULES**

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZGRP5\_MAINPROG*  
*\*&---------------------------------------------------------------------\**  
  
loop at gt\_input into gwa\_input.  
  
  
  
ls\_cont\_hdr\_in-doc\_type = gwa\_input-contract\_type.      *"'KM'. " Contract Type( CQ->KM)*  
ls\_cont\_hdr\_in-purch\_no\_c = 'PO\_NUM'. *"Purchase Order*  
ls\_cont\_hdr\_in-sales\_org = gwa\_input-vkorg.  *"'1710'. " Sales Organization*  
ls\_cont\_hdr\_in-distr\_chan = gwa\_input-vtweg.    *"'10'. " Distribution Channel*  
ls\_cont\_hdr\_in-division = gwa\_input-spart.  *"'00'. " Division*  
ls\_cont\_hdr\_in-ct\_valid\_f = gwa\_input-valid\_from.   *"'20240731'. " Contract Valid From date*  
ls\_cont\_hdr\_in-ct\_valid\_t = gwa\_input-valid\_to.   *"'20240831'. " Contract valid to date*  
  
ls\_cont\_hdr\_inx-doc\_type = 'X'.  
ls\_cont\_hdr\_inx-purch\_no\_c = 'x'.  
ls\_cont\_hdr\_inx-sales\_org = 'X'.  
ls\_cont\_hdr\_inx-distr\_chan = 'X'.  
ls\_cont\_hdr\_inx-division = 'X'.  
ls\_cont\_hdr\_inx-ct\_valid\_f = 'X'.  
ls\_cont\_hdr\_inx-ct\_valid\_t = 'X'.  
  
  
*\* Prepare Contract Item Data*  
APPEND INITIAL LINE TO lt\_item\_in ASSIGNING FIELD-SYMBOL(<fs\_item\_in>).  
<fs\_item\_in>-itm\_number = '000010'.  
<fs\_item\_in>-material = gwa\_input-matnr. *"'MZ-TG-Y120'.*  
<fs\_item\_in>-plant = ''.  
<fs\_item\_in>-ship\_point = ''.  
<fs\_item\_in>-target\_qty = gwa\_input-quantity.  *"'10'.*  
<fs\_item\_in>-target\_qu =  'ST'. *" PC->ST*  
  
APPEND INITIAL LINE TO lt\_item\_inx ASSIGNING FIELD-SYMBOL(<fs\_item\_inx>).  
<fs\_item\_inx>-itm\_number = 'X'.  
<fs\_item\_inx>-material = 'X'.  
<fs\_item\_inx>-plant = 'X'.  
<fs\_item\_inx>-ship\_point = 'X'.  
<fs\_item\_inx>-target\_qty = 'X'.  
<fs\_item\_inx>-target\_qu = 'X'.  
APPEND INITIAL LINE TO lt\_partner ASSIGNING FIELD-SYMBOL(<fs\_partner>).  
<fs\_partner>-partn\_role = 'AG'. *" Sold to Party*  
<fs\_partner>-partn\_numb = gwa\_input-sold\_to.  *"'USCU\_L10'.*  
  
CALL FUNCTION 'BAPI\_CONTRACT\_CREATEFROMDATA'  
EXPORTING  
  
contract\_header\_in = ls\_cont\_hdr\_in  
contract\_header\_inx = ls\_cont\_hdr\_inx  
  
IMPORTING  
salesdocument = lv\_salesdoc  
TABLES  
return = lt\_return  
contract\_items\_in = lt\_item\_in  
contract\_items\_inx = lt\_item\_inx  
contract\_partners = lt\_partner  
  
.  
  
CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
EXPORTING  
wait = 'X'.  
  
LOOP AT lt\_return INTO DATA(WA\_RETURN).  
    if wa\_return-type = 'S'.  
      LV\_VAR1 = 'SUCCESS'.  
      ELSE.  
        LV\_VAR1 = 'ERROR'.  
     ENDIF.  
  
    gwa\_input-file\_name = p\_file.  
    gwa\_input-contract\_num = lv\_salesdoc.  
    gwa\_input-erdat = SY-DATUM.  
    gwa\_input-status =  LV\_VAR1 .  
    gwa\_input-username = SY-UNAME.  
    gwa\_input-comment1 = WA\_return-message.  
    append gwa\_input to it\_tab22.  
 ENDLOOP.  
  
SET UPDATE TASK LOCAL.  
 modify zg05\_table1 from gwa\_input.  
  
 endloop.

**INCLUDE FOR DISPLAYING THE OOALV SCREEN**

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZGRP\_OOALVSCREEN*  
*\*&---------------------------------------------------------------------\**  
wa\_fieldcat-col\_pos = '1'.  
wa\_fieldcat-coltext = 'FILENAME'.  
wa\_fieldcat-fieldname = 'FILE\_NAME'.  
wa\_fieldcat-tabname  = 'ZG05\_TABLE1'.  
  
APPEND wa\_fieldcat TO lt\_fieldcat.  
CLEAR wa\_fieldcat.  
  
wa\_fieldcat-col\_pos = '2'.  
wa\_fieldcat-coltext = 'CONTRACT NUMBER'.  
wa\_fieldcat-fieldname = 'CONTRACT\_NUM'.  
wa\_fieldcat-tabname  = 'ZG05\_TABLE1'.  
  
APPEND wa\_fieldcat TO lt\_fieldcat.  
CLEAR wa\_fieldcat.  
  
wa\_fieldcat-col\_pos = '3'.  
wa\_fieldcat-coltext = 'STATUS'.  
wa\_fieldcat-fieldname = 'STATUS'.  
wa\_fieldcat-tabname  = 'ZG05\_TABLE1'.  
  
APPEND wa\_fieldcat TO lt\_fieldcat.  
CLEAR wa\_fieldcat.  
  
wa\_fieldcat-col\_pos = '4'.  
wa\_fieldcat-coltext = 'COMMENT'.  
wa\_fieldcat-fieldname = 'COMMENT1'.  
wa\_fieldcat-tabname  = 'ZG05\_TABLE1'.  
  
APPEND wa\_fieldcat TO lt\_fieldcat.  
CLEAR wa\_fieldcat.  
  
  
SELECT file\_name , contract\_num , status , comment1 FROM zg05\_table1 INTO TABLE @DATA(gt\_tab)  
  FOR ALL ENTRIES IN @it\_tab22  
  WHERE bstnk = @it\_tab22-bstnk .  
cl\_demo\_output=>display( gt\_tab ).  
  
CALL SCREEN 100.

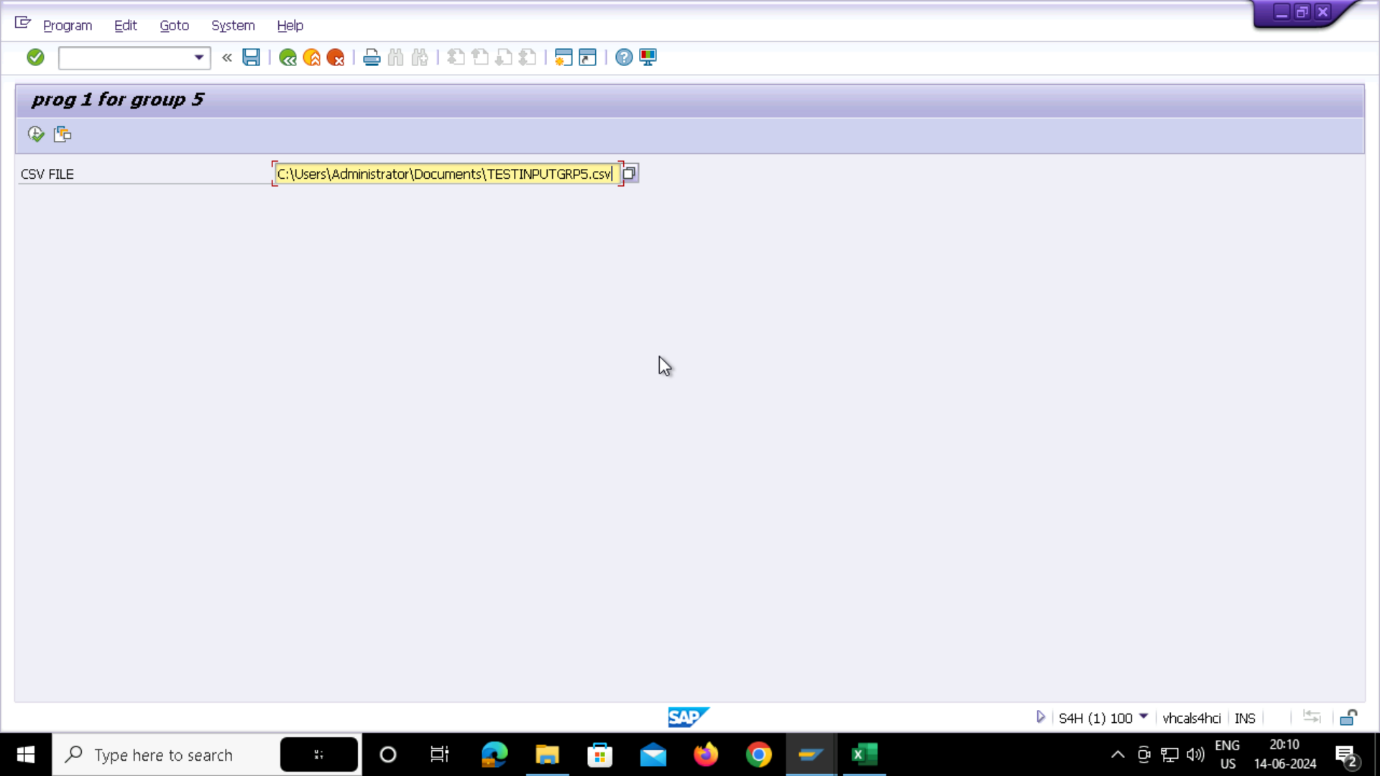
**INCLUDE FOR DISPLAYING THE CONTRACT DETAILS IN VA43**

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZGRP5\_GETCURSOR*  
*\*&---------------------------------------------------------------------\**  
  
  
*\*&---------------------------------------------------------------------\**  
*\*&  class defination for the double-click*  
*\*&---------------------------------------------------------------------\**  
  
DATA : FNAM(15),  
       FVAL(15).  
  
class lcl\_alv\_inter definition.  
  public section.  
    methods : handle\_double\_click for event double\_click of cl\_gui\_alv\_grid  importing e\_row.  
  
endclass.  
  
  
*\*&---------------------------------------------------------------------\**  
*\*&   class implementation of double click*  
*\*&---------------------------------------------------------------------\**  
  
class lcl\_alv\_inter implementation.  
*\*-------------------------------------------------------------------------------------------------\**  
  method handle\_double\_click.  
  
  
    CALL TRANSACTION 'VA43' AND SKIP FIRST SCREEN.  
  
  ENDMETHOD.  
  
ENDCLASS.

**INCLUDE FOR MODULE DEFINATIONS FOR OOALV OUTPUT**

*\*&---------------------------------------------------------------------\**  
*\*& Include          ZGRP5\_ALV*  
*\*&---------------------------------------------------------------------\**  
  
  
*\*&---------------------------------------------------------------------\**  
*\*& Module OOALV\_REP OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
MODULE ooalv\_rep OUTPUT.  
  SET PF-STATUS space.  
  SET TITLEBAR 'xxx'.  
  
  
  CREATE OBJECT i\_container  
    EXPORTING  
      container\_name = 'CTN1'.  
  IF sy-subrc <> 0.  
*\*       MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO*  
*\*              WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.*  
  ENDIF.  
  
  CREATE OBJECT i\_grid  
    EXPORTING  
      i\_parent = i\_container.  
  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 METHOD i\_grid->set\_table\_for\_first\_display  
    EXPORTING  
      i\_save          = 'X'  
      i\_default       = 'X'  
*\**  
    CHANGING  
      it\_outtab       = gt\_tab  
      it\_fieldcatalog = lt\_fieldcat  
*\**  
    .  
  IF sy-subrc <> 0.  
*\*                 Implement suitable error handling here*  
  ENDIF.  
  
  DATA i\_handler TYPE REF TO lcl\_alv\_inter.  
  
  CREATE OBJECT i\_handler.  
  
  SET HANDLER : i\_handler->handle\_double\_click FOR i\_grid.  
  
  
  
  
ENDMODULE.  
  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  USER\_COMMAND\_0100  INPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module user\_command\_0100 input.  
  DATA ok\_code type sy-ucomm.  
  CASE ok\_code.  
    WHEN 'BACK'.  
      LEAVE TO SCREEN 0.  
  ENDCASE.  
  
endmodule.

**INPUT SELECTION SCREEN:**



A screenshot of a computer

Description automatically generated

**GENERATING THE OOALV OUTPUT LIKE THIS:**

IF THE CONTRACT NUMBER IS CREATED SUCESSFULLY IT WILL GIVE THE OUTPUT LIKE WITH STATUS “SUCCESS” OTHERWISE IT WILL GIVE THE STATUS “ERROR”.

A screenshot of a computer

Description automatically generated

**WHENEVER WE DOUBLE CLICK ON THE PARTICULAR SUCESSFULLY CREATED CONTRACT NUMBER IT IS GOING TO OPEN THE VA43(DISPLAY) FOR THE PARTICULAR CONTRACT NUMBER WHICH IS CREATED.**

A screenshot of a computer

Description automatically generated