Program Name: ZABAP_EX16GBLCLASS_PDC6

Description: Program Implementation of Global Class (Create PO and SO number, Display in ALV Format, With Clickable function on each cell in ALV to go to TCODE ME23N)

Source Code:

```
*&---------------------------------------------------------------------*
*& Report ZABAP_EX16GBLCLASS_PDC6
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT zabap_ex16gblclass_pdc6.

DATA: gt_poheader  TYPE zcl_pdc6_ex16=>ty_header,
      gs_poheader  TYPE zcl_pdc6_ex16=>ty_h,
      gt_poitm     TYPE zcl_pdc6_ex16=>ty_itm,
      gs_poitm     TYPE zcl_pdc6_ex16=>ty_i,
      gt_return    TYPE zcl_pdc6_ex16=>ty_return,
      gv_ebeln     TYPE bapimepoheader-po_number,
      gt_po        TYPE zcl_pdc6_ex16=>ty_po,

      "FOR SO
      gt_soheader  TYPE zcl_pdc6_ex16=>ty_soheader,
      gt_soitm     TYPE zcl_pdc6_ex16=>ty_soitm,
      gv_vbeln     TYPE bapivbeln-vbeln,
      gt_so        TYPE zcl_pdc6_ex16=>ty_so,


      "this is for ALV class declaration.
      gcl_salv_tbl TYPE REF TO cl_salv_table.

INITIALIZATION.
  FREE: gt_poheader, gs_poheader, gt_poitm, gs_poitm, gt_return, gv_ebeln, gt
_po, gt_soheader,
        gt_soitm, gv_vbeln, gt_so.

SELECTION-SCREEN BEGIN OF BLOCK frame0 WITH FRAME TITLE TEXT-man.

SELECTION-SCREEN BEGIN OF BLOCK frame1 WITH FRAME TITLE TEXT-btn.
SELECTION-SCREEN PUSHBUTTON 10(13) b_so USER-COMMAND som.
SELECTION-SCREEN PUSHBUTTON 25(15) b_po USER-COMMAND pom.
SELECTION-SCREEN END OF BLOCK frame1.

SELECTION-SCREEN BEGIN OF BLOCK frame2 WITH FRAME TITLE TEXT-rbt.
PARAMETERS: rbsi RADIOBUTTON GROUP rbsl USER-COMMAND test DEFAULT 'X',
            rbpo RADIOBUTTON GROUP rbsl,
            rbso RADIOBUTTON GROUP rbsl.
SELECTION-SCREEN END OF BLOCK frame2.
```

```
SELECTION-SCREEN BEGIN OF BLOCK frame3 WITH FRAME TITLE TEXT-fld.
SELECTION-SCREEN BEGIN OF BLOCK frame4 WITH FRAME TITLE TEXT-hdr.
PARAMETERS: p_bsart TYPE ekko-bsart DEFAULT 'NB'  MODIF ID pop,
            p_org   TYPE ekko-ekorg MODIF ID pop,
            p_grp   TYPE ekko-ekgrp MODIF ID pop,
            p_bukrs TYPE ekko-bukrs MODIF ID pop,
            p_vndr  TYPE ekko-lifnr MODIF ID pop.
SELECTION-SCREEN END OF BLOCK frame4.

SELECTION-SCREEN BEGIN OF BLOCK frame5 WITH FRAME TITLE TEXT-itm.
PARAMETERS: p_matnr TYPE ekpo-matnr MODIF ID pop,
            p_menge TYPE ekpo-menge MODIF ID pop,
            p_netpr TYPE ekpo-netpr MODIF ID pop,
            p_werks TYPE ekpo-werks MODIF ID pop.
SELECTION-SCREEN END OF BLOCK frame5.
SELECTION-SCREEN END OF BLOCK frame3.

SELECTION-SCREEN END OF BLOCK frame0.

AT SELECTION-SCREEN OUTPUT.
  b_so = TEXT-001. "SO
  b_po = TEXT-002. "PO

  LOOP AT SCREEN.
    IF rbsi EQ abap_true AND screen-group1 EQ TEXT-pop.
      screen-active = 1.
      MODIFY SCREEN.
      CONTINUE.
    ELSEIF ( rbpo EQ abap_true OR rbso EQ abap_true ) AND screen-
group1 EQ TEXT-pop.
      FREE: p_org, p_grp, p_bukrs, p_vndr, p_matnr, p_menge, p_netpr, p_werks
.
      screen-active = 0.
      MODIFY SCREEN.
      CONTINUE.
    ENDIF.
  ENDLOOP.

AT SELECTION-SCREEN.
  CASE sy-ucomm.
    WHEN TEXT-som.
      CALL METHOD zcl_pdc6_ex16=>download_template
        EXPORTING
          lv_potmpl_mark = ''.
    WHEN TEXT-pom.
      CALL METHOD zcl_pdc6_ex16=>download_template
        EXPORTING
          lv_potmpl_mark = 'X'.
  ENDCASE.

START-OF-SELECTION.

END-OF-SELECTION.

  CASE abap_true.
    WHEN rbsi.
      IF p_bsart IS INITIAL OR p_org IS INITIAL OR p_grp IS INITIAL OR p_bukr
s IS INITIAL OR p_vndr IS INITIAL OR
         p_matnr IS INITIAL OR p_menge IS INITIAL OR p_netpr IS INITIAL OR p_
werks IS INITIAL.
        MESSAGE: TEXT-003 TYPE 'S'.
      ELSE.
        FREE: gt_poheader, gs_poheader.

        gs_poheader-keylink = TEXT-nm1. "Start of passing PO header value.
        gs_poheader-doctyp = p_bsart.
        gs_poheader-comp_code = p_bukrs.
        gs_poheader-purch_org = p_org.
        gs_poheader-purch_grp = p_grp.
        gs_poheader-vendor = p_vndr.

        APPEND gs_poheader TO gt_poheader.

        gs_poitm-keylink = TEXT-nm1. "Start of passing PO item value
        gs_poitm-matnr = p_matnr.
        gs_poitm-qty = p_menge.
        gs_poitm-price   = p_netpr.
        gs_poitm-plant = p_werks.

        APPEND gs_poitm TO gt_poitm.

        CALL METHOD zcl_pdc6_ex16=>create_po_no
          EXPORTING
            lt_poitm  = gt_poitm
            lt_header = gt_poheader
          IMPORTING
            lt_return = gt_return
            lv_ebeln  = gv_ebeln
            lt_po     = gt_po.
        IF sy-subrc EQ 0.
          CALL METHOD cl_salv_table=>factory
            IMPORTING
              r_salv_table = gcl_salv_tbl
            CHANGING
              t_table      = gt_po.
          IF sy-subrc EQ 0.
            gcl_salv_tbl->get_functions( )->set_all( abap_true ).
            gcl_salv_tbl->get_columns( )->set_optimize( abap_true ).

            zcl_pdc6_ex16=>lt_poalv = gt_po.
            SET HANDLER zcl_pdc6_ex16=>click_me FOR gcl_salv_tbl-
>get_event( ).
            SET HANDLER zcl_pdc6_ex16=>clickme_func FOR gcl_salv_tbl-
>get_event( ).

            gcl_salv_tbl->display( ).
          ENDIF.
        ENDIF.
      ENDIF.

    WHEN rbpo.
      CALL METHOD zcl_pdc6_ex16=>upload_file
        EXPORTING
          lv_po_mark = abap_true
        IMPORTING
          lt_header  = gt_poheader
          lt_itm     = gt_poitm.
      IF gt_poheader[] IS NOT INITIAL AND gt_poitm[] IS NOT INITIAL.
        CALL METHOD zcl_pdc6_ex16=>create_po_no
          EXPORTING
            lt_poitm  = gt_poitm
            lt_header = gt_poheader
          IMPORTING
            lt_return = gt_return
            lv_ebeln  = gv_ebeln
            lt_po     = gt_po.
        IF sy-subrc EQ 0.
          CALL METHOD cl_salv_table=>factory
            IMPORTING
              r_salv_table = gcl_salv_tbl
            CHANGING
              t_table      = gt_po.
          IF sy-subrc EQ 0.
            gcl_salv_tbl->get_functions( )->set_all( abap_true ).
            gcl_salv_tbl->get_columns( )->set_optimize( abap_true ).

            zcl_pdc6_ex16=>lv_so_mark = space.
            zcl_pdc6_ex16=>lt_poalv = gt_po.
            SET HANDLER zcl_pdc6_ex16=>click_me FOR gcl_salv_tbl-
>get_event( ).
            SET HANDLER zcl_pdc6_ex16=>clickme_func FOR gcl_salv_tbl-
>get_event( ).

            gcl_salv_tbl->display( ).
          ENDIF.
        ENDIF.
      ENDIF.
    WHEN rbso.
      CALL METHOD zcl_pdc6_ex16=>upload_file
        EXPORTING
          lv_po_mark  = space
        IMPORTING
          lt_soitm    = gt_soitm
          lt_soheader = gt_soheader.
      IF gt_soheader[] IS NOT INITIAL AND gt_soitm[] IS NOT INITIAL.
        CALL METHOD zcl_pdc6_ex16=>create_so_no
          EXPORTING
            lt_so_itm   = gt_soitm
            lt_soheader = gt_soheader
          IMPORTING
            lt_soreturn = gt_return
            lv_sovbeln  = gv_vbeln
            lt_so       = gt_so.
        IF sy-subrc EQ 0.
          CALL METHOD cl_salv_table=>factory
            IMPORTING
              r_salv_table = gcl_salv_tbl
            CHANGING
              t_table      = gt_so.
          IF sy-subrc EQ 0.
            gcl_salv_tbl->get_functions( )->set_all( abap_true ).
            gcl_salv_tbl->get_columns( )->set_optimize( abap_true ).

            zcl_pdc6_ex16=>lv_so_mark = abap_true.
            zcl_pdc6_ex16=>lt_soalv = gt_so.
            SET HANDLER zcl_pdc6_ex16=>click_me FOR gcl_salv_tbl-
>get_event( ).
            SET HANDLER zcl_pdc6_ex16=>clickme_func FOR gcl_salv_tbl-
>get_event( ).

            gcl_salv_tbl->display( ).
          ENDIF.
        ENDIF.
      ENDIF.

  ENDCASE.

  FREE: gt_poheader, gt_poitm, gt_soheader, gt_soitm, gt_return.
```

Text Elements: (Text Symbols)

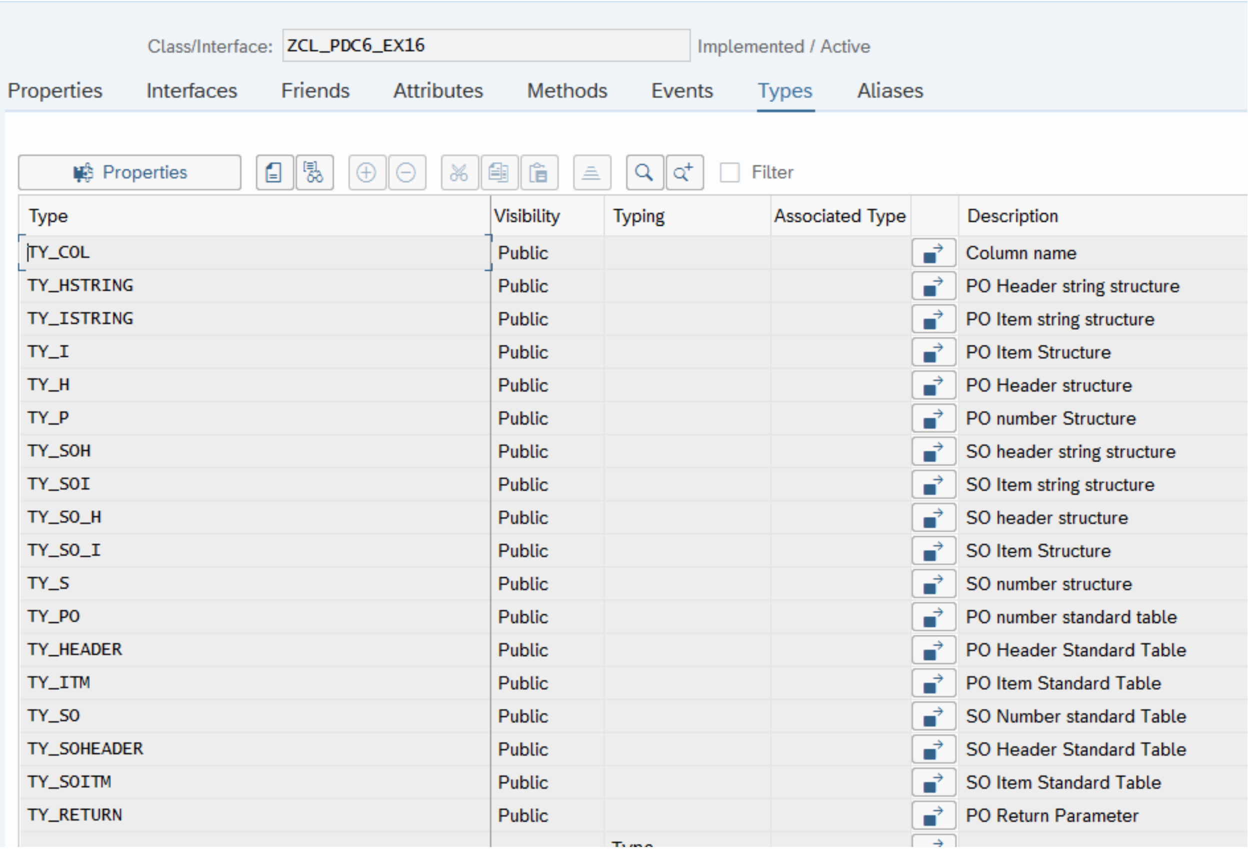
SYM	TEXT	LENGTH	MAX.
001	Sales Order	11	21
002	Purchase Order	14	24
003	Please fillout all fields!	26	52
004	SO Template Downloaded, After Editing your template make sure to save it to an updated version of excel.	104	132
005	PO Template Downloaded, After Editing your template make sure to save it to an updated version of excel.	104	132
BTN	Download XLSX Template	22	44
FLD	Purchase Order Inputs	21	42
HDR	PO Header	9	19
ITM	PO Items	8	18
NM1	1	1	11
POM	POM	3	13
POP	POP	3	13
RBT	Insert Items	12	22
SOM	SOM	3	13

Selection Texts:

NAME	TEXT
P_BSART	Document Type
P_BUKRS	Company Code
P_GRP	Purch Group
P_MATNR	Material
P_MENGE	Quantity
P_NETPR	Net Price
P_ORG	Purch. Org.
P_VNDR	Vendor
P_WERKS	Plant
RBPO	Multiple PO Items
RBSI	Single PO Items
RBSO	Multiple SO Items

Class Name: ZCL_PDC6_EX16

Types:



Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Properties Interfaces Friends Attributes Methods Events Types Aliases

Properties □ Filter

Type	Visibility	Typing	Associated Type	Description
TY_COL	Public			Column name
TY_HSTRING	Public			PO Header string structure
TY_ISTRING	Public			PO Item string structure
TY_I	Public			PO Item Structure
TY_H	Public			PO Header structure
TY_P	Public			PO number Structure
TY_SOH	Public			SO header string structure
TY_SOI	Public			SO Item string structure
TY_SO_H	Public			SO header structure
TY_SO_I	Public			SO Item Structure
TY_S	Public			SO number structure
TY_PO	Public			PO number standard table
TY_HEADER	Public			PO Header Standard Table
TY_ITM	Public			PO Item Standard Table
TY_SO	Public			SO Number standard Table
TY_SOHEADER	Public			SO Header Standard Table
TY_SOITM	Public			SO Item Standard Table
TY_RETURN	Public			PO Return Parameter

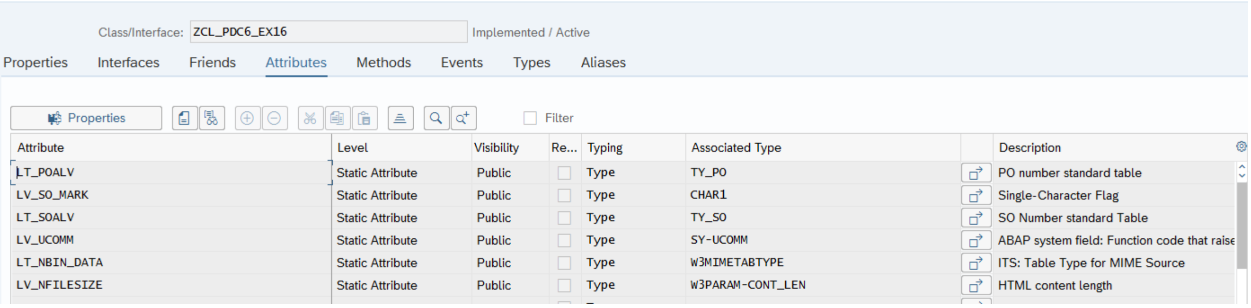
Types Source Code:

```
class ZCL_PDC6_EX16 definition
  public
  final
  create public .

public section.

  types:
    BEGIN OF ty_col,
        colname TYPE string,
      END OF ty_col .
  types:
      "START OF STRUCTURES FOR PURCHASE ORDER.
    BEGIN OF ty_hstring,
        keylink   TYPE string,
        doctyp    TYPE string,
        vendor    TYPE string,
        purch_org TYPE string,
        purch_grp TYPE string,
        comp_code TYPE string,
      END OF ty_hstring .
  types:
```

```
  BEGIN OF ty_istring,
      keylink TYPE string,
      matnr   TYPE string,
      qty     TYPE string,
      price   TYPE string,
      plant   TYPE string,
    END OF ty_istring .
types:
  BEGIN OF ty_i,
      keylink TYPE i,
      matnr   TYPE mepo1211-ematn,
      qty     TYPE mepo1211-menge,
      price   TYPE mepo1211-netpr,
      plant   TYPE mepo1211-name1,
    END OF ty_i .
types:
  BEGIN OF ty_h,
      keylink   TYPE i,
      doctyp    TYPE ekko-bsart,
      vendor    TYPE lifnr,
      purch_org TYPE mepo1222-ekorg,
      purch_grp TYPE mepo1222-ekgrp,
      comp_code TYPE mepo1222-bukrs,
    END OF ty_h .
types:
  BEGIN OF ty_p,
      keylink TYPE zabap_keylink_de,
      pono    TYPE ebeln,
    END OF ty_p .
types:
    "END OF STRUCTURES FOR PURCHASE ORDER.
    "START OF STRUCTURES FOR SALES ORDER.
  BEGIN OF ty_soh,
      keylink   TYPE string, " test
      doctyp    TYPE string,
      salesorg  TYPE string,
      dchannel  TYPE string,
      division  TYPE string,
      custporef TYPE string,
      prtnr1    TYPE string,
      prtnr2    TYPE string,
    END OF ty_soh .
types:
  BEGIN OF ty_soi,
      keylink TYPE string, " test
      matr    TYPE string,
      qty     TYPE string,
    END OF ty_soi .
types:
  BEGIN OF ty_so_h,
      keylink   TYPE i,
      doctyp    TYPE auart,
      salesorg  TYPE vkorg,
      dchannel  TYPE vtweg,
      division  TYPE spart,
      custporef TYPE bstnk,
      prtnr1    TYPE kunag,
      prtnr2    TYPE kunag,
    END OF ty_so_h .
types:
  BEGIN OF ty_so_i,
      keylink TYPE i, " test
      matr    TYPE matnr,
      qty     TYPE dzmeng,
    END OF ty_so_i .
types:
  BEGIN OF ty_s,
      keylink TYPE zabap_keylink_de,
      sono    TYPE vbeln,
    END OF ty_s .
types:
  "END OF STRUCTURES FOR SALES ORDER.
  "START OF STANDARD TABLE FOR PURCHASE ORDER.
  ty_po TYPE STANDARD TABLE OF ty_p .
types:
  ty_header TYPE STANDARD TABLE OF ty_h .
types:
  ty_itm TYPE STANDARD TABLE OF ty_i .
types:
  "END OF STANDARD TABLE FOR PURCHASE ORDER.
  "START OF STANDARD TABLE FOR SALES ORDER.
  ty_so TYPE STANDARD TABLE OF ty_s .
types:
  ty_soheader TYPE STANDARD TABLE OF ty_so_h .
types:
  ty_soitm TYPE STANDARD TABLE OF ty_so_i .
types:
  "END OF STANDARD TABLE FOR SALES ORDER.
  ty_return TYPE STANDARD TABLE OF bapiret2 .

class-data LT_POALV type TY_PO .
class-data LV_SO_MARK type CHAR1 .
class-data LT_SOALV type TY_SO .
class-data LV_UCOMM type SY-UCOMM .
class-data LT_NBIN_DATA type W3MIMETABTYPE .
class-data LV_NFILESIZE type W3PARAM-CONT_LEN .

class-methods CREATE_PO_NO
```

```
    importing
      value(LT_POITM) type TY_ITM
      value(LT_HEADER) type TY_HEADER
    exporting
      value(LT_RETURN) type TY_RETURN
      value(LV_EBELN) type BAPIMEPOHEADER-PO_NUMBER
      value(LT_PO) type TY_PO .
  class-methods UPLOAD_FILE
    importing
      value(LV_PO_MARK) type CHAR1
    exporting
      value(LT_HEADER) type TY_HEADER
      value(LT_ITM) type TY_ITM
      value(LT_SOITM) type TY_SOITM
      value(LT_SOHEADER) type TY_SOHEADER .
  class-methods CREATE_SO_NO
    importing
      value(LT_SO_ITM) type TY_SOITM
      value(LT_SOHEADER) type TY_SOHEADER
    exporting
      value(LT_SORETURN) type TY_RETURN
      value(LV_SOVBELN) type BAPIVBELN-VBELN
      value(LT_SO) type TY_SO .
  class-methods CLICK_ME
    for event DOUBLE_CLICK of CL_SALV_EVENTS_TABLE
    importing
      !ROW
      !COLUMN .
  class-methods CLICKME_FUNC
    for event ADDED_FUNCTION of CL_SALV_EVENTS_TABLE
    importing
      !E_SALV_FUNCTION .
  class-methods DOWNLOAD_TEMPLATE
    importing
      value(LV_POTMPL_MARK) type CHAR1
      value(LV_EXP_UPLOAD) type CHAR1 optional .
```

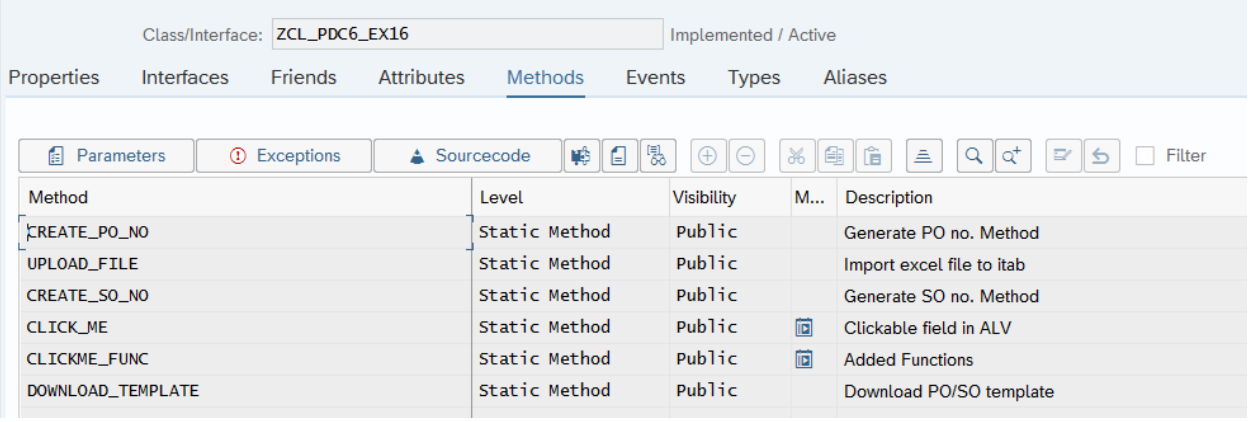
Attributes:



Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Attribute	Level	Visibility	Re...	Typing	Associated Type		Description
LT_POALV	Static Attribute	Public	□	Type	TY_PO		PO number standard table
LV_SO_MARK	Static Attribute	Public	□	Type	CHAR1		Single-Character Flag
LT_SOALV	Static Attribute	Public	□	Type	TY_SO		SO Number standard Table
LV_UCOMM	Static Attribute	Public	□	Type	SY-UCOMM		ABAP system field: Function code that raise
LT_NBIN_DATA	Static Attribute	Public	□	Type	W3MIMETABTYPE		ITS: Table Type for MIME Source
LV_NFILESIZE	Static Attribute	Public	□	Type	W3PARAM-CONT_LEN		HTML content length

Methods:

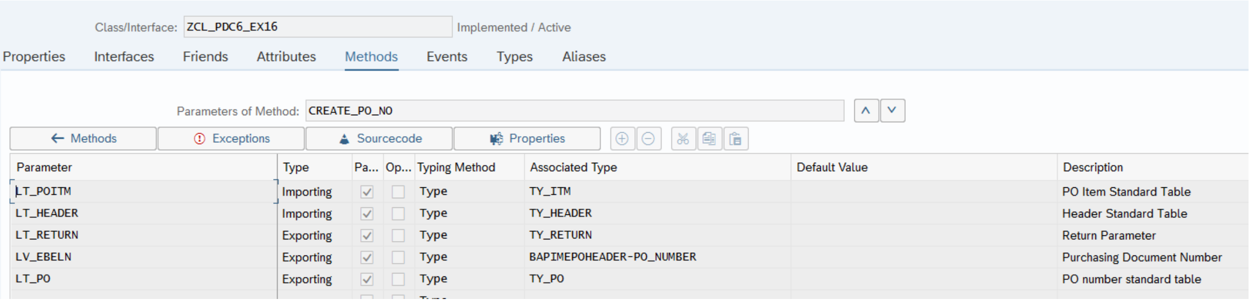


Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Method	Level	Visibility	M...	Description
CREATE_PO_NO	Static Method	Public		Generate PO no. Method
UPLOAD_FILE	Static Method	Public		Import excel file to itab
CREATE_SO_NO	Static Method	Public		Generate SO no. Method
CLICK_ME	Static Method	Public	▣	Clickable field in ALV
CLICKME_FUNC	Static Method	Public	▣	Added Functions
DOWNLOAD_TEMPLATE	Static Method	Public		Download PO/SO template

Method Name: CREATE_PO_NO

Parameters:



Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Parameters of Method: CREATE_PO_NO

Parameter	Type	Pa...	Op...	Typing Method	Associated Type	Default Value	Description
LT_POITM	Importing	☑	□	Type	TY_ITM		PO Item Standard Table
LT_HEADER	Importing	☑	□	Type	TY_HEADER		Header Standard Table
LT_RETURN	Exporting	☑	□	Type	TY_RETURN		Return Parameter
LV_EBELN	Exporting	☑	□	Type	BAPIMEPOHEADER-PO_NUMBER		Purchasing Document Number
LT_PO	Exporting	☑	□	Type	TY_PO		PO number standard table

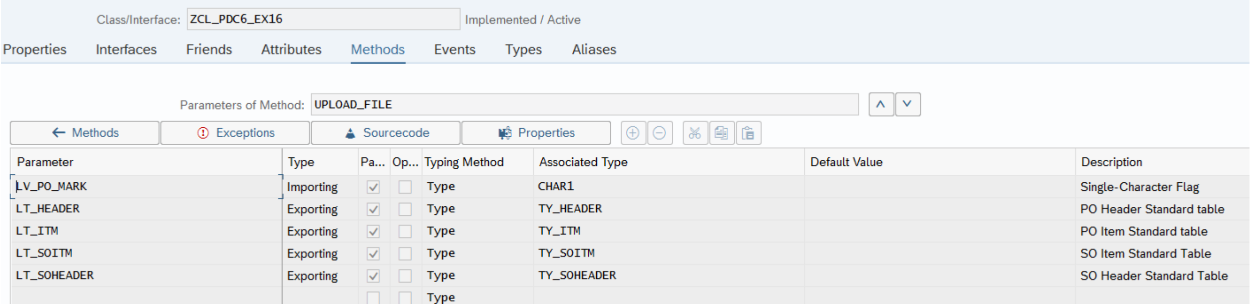
Source Code:

```
METHOD create_po_no.
  DATA: lv_poitem_no TYPE ebelp,
        ls_poheader  TYPE bapimepoheader,
        ls_poheaderx TYPE bapimepoheaderx,
        lt_poitem    TYPE STANDARD TABLE OF bapimepoitem,
        lt_poitemx   TYPE STANDARD TABLE OF bapimepoitemx,
        ls_poitem    TYPE bapimepoitem,
        ls_poitemx   TYPE bapimepoitemx,
        ls_po        TYPE ty_p,
        ls_return    TYPE bapiret2.

  CONSTANTS: lc_e_mark TYPE char1 VALUE 'E'.

  LOOP AT lt_header INTO DATA(ls_header) WHERE keylink NE space.
    ls_poheader-doc_type = ls_header-doctyp.
    ls_poheader-comp_code = ls_header-comp_code.
    ls_poheader-purch_org = ls_header-purch_org.
```

```
      ls_poheader-pur_group = ls_header-purch_grp.
      ls_poheader-vendor = ls_header-vendor.

      ls_poheaderx-doc_type = abap_true.
      ls_poheaderx-comp_code = abap_true.
      ls_poheaderx-purch_org = abap_true.
      ls_poheaderx-pur_group = abap_true.
      ls_poheaderx-vendor = abap_true.

      LOOP AT lt_poitm INTO DATA(ls_poitm) WHERE keylink EQ ls_header-
keylink AND matnr NE space.

        SELECT SINGLE maktx FROM makt INTO @DATA(lv_sdesc) WHERE matnr EQ @ls
_poitm-matnr AND spras EQ @lc_e_mark.

        lv_poitem_no = lv_poitem_no + 1.

        ls_poitem-po_item = lv_poitem_no.
        ls_poitem-material = ls_poitm-matnr.
        ls_poitem-short_text = lv_sdesc.
        ls_poitem-plant = ls_poitm-plant.
        ls_poitem-quantity = ls_poitm-qty.
        ls_poitem-net_price = ls_poitm-price.

        APPEND ls_poitem TO lt_poitem.
        FREE ls_poitem.

        ls_poitemx-po_item = lv_poitem_no.
        ls_poitemx-po_itemx = abap_true.
        ls_poitemx-material = abap_true.
        ls_poitemx-short_text = abap_true.
        ls_poitemx-plant = abap_true.
        ls_poitemx-quantity = abap_true.
        ls_poitemx-net_price = abap_true.

        APPEND ls_poitemx TO lt_poitemx.
        FREE: ls_poitemx, ls_poitm.
      ENDLOOP.

      CALL FUNCTION 'BAPI_PO_CREATE1'
        EXPORTING
          poheader         = ls_poheader
          poheaderx        = ls_poheaderx
          no_price_from_po = abap_true
        IMPORTING
          exppurchaseorder = lv_ebeln
        TABLES
          return           = lt_return
          poitem           = lt_poitem
          poitemx          = lt_poitemx.

      IF lv_ebeln IS NOT INITIAL.
        CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
          EXPORTING
            wait = abap_true.
        IF sy-subrc EQ 0.
"START OF ADD BY JIMBED
SORT LT_PO BY KEYLINK ASCENDING.
"END OF ADD BY JIMBED
*            MESSAGE lv_ebeln TYPE 'I'.
          ls_po-keylink = ls_header-keylink.
          ls_po-pono = lv_ebeln.

          APPEND ls_po TO lt_po.
        ENDIF.
      ENDIF.

"START OF add BY JIMBED
 "NEW CODE
IF lt_return IS NOT INITIAL.
*      CALL FUNCTION 'C14ALD_BAPIRET2_SHOW' "NEW FM FOR DISPLAY ERRORS
*        TABLES
*          i_bapiret2_tab       = lt_return.
      CALL FUNCTION 'ZPDC7_ERRORS_MSG_SHOW'
        TABLES
          i_bapiret2_tab       = LT_RETURN
                .

ENDIF.
"END OF CHANGE BY JIMBED

      FREE: ls_header, ls_poheader, ls_poheaderx, ls_po, lt_poitem, lt_poitem
x, lt_return.
    ENDLOOP.
  ENDMETHOD.
```

Method Name: UPLOAD_FILE

Parameters:



Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Parameters of Method: UPLOAD_FILE

Parameter	Type	Pa...	Op...	Typing Method	Associated Type	Default Value	Description
LV_PO_MARK	Importing	☑	□	Type	CHAR1		Single-Character Flag
LT_HEADER	Exporting	☑	□	Type	TY_HEADER		PO Header Standard table
LT_ITM	Exporting	☑	□	Type	TY_ITM		PO Item Standard table
LT_SOITM	Exporting	☑	□	Type	TY_SOITM		SO Item Standard Table
LT_SOHEADER	Exporting	☑	□	Type	TY_SOHEADER		SO Header Standard Table
		□	□	Type			

Source Code:

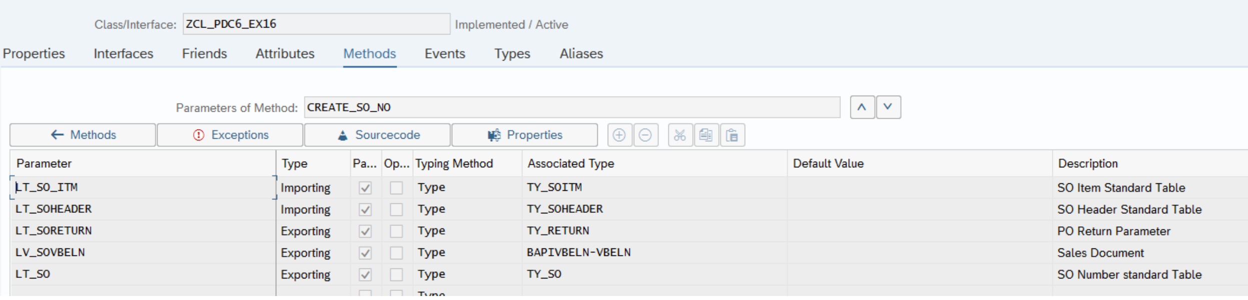
```
  METHOD upload_file.
"START OF ADD BY JIMBED
    CONSTANTS: GC_E TYPE CHAR1 VALUE 'E',
               GC_S TYPE CHAR1 VALUE 'S',
               GC_0 TYPE CHAR1 VALUE 0.
"END OF ADD BY JIMBED

    DATA: "FOR GENERATING PO. Number.
      lt_s_header         TYPE STANDARD TABLE OF ty_hstring,
      ls_header           TYPE ty_h,
      lt_s_itm            TYPE STANDARD TABLE OF ty_istring,
      ls_itm              TYPE ty_i,
      "FOR GENERATING SO. Number.
      lt_s_soheader       TYPE STANDARD TABLE OF ty_soh,
      lt_s_soitm          TYPE STANDARD TABLE OF ty_soi,
      ls_soheader         TYPE ty_so_h,
      ls_soitm            TYPE ty_so_i,

      "DECLARATION FOR WORKSHEET CLASS.
      lv_rc               TYPE i, "this is for row count...
      lt_files            TYPE filetable,
      lv_action           TYPE i,

      lv_filesize         TYPE w3param-cont_len,
      lt_bin_data         TYPE w3mimetabtype,

      "this is to get first the worksheet name.
      lt_worksheet_names TYPE if_fdt_doc_spreadsheet=>t_worksheet_names.

    "this is for file open dialog.
    cl_gui_frontend_services=>file_open_dialog(
      EXPORTING
        file_filter = |xlsx (*.xlsx)\|*.xlsx\|{ cl_gui_frontend_services=>fil
etype_all }|
      CHANGING
        file_table  = lt_files
        rc          = lv_rc
        user_action = lv_action
    ).

    IF lv_action = cl_gui_frontend_services=>action_ok.
      IF lines( lt_files ) > 0.

        "this is for read file and GUI upload.
        cl_gui_frontend_services=>gui_upload(
          EXPORTING
            filename = |{ lt_files[ 1 ]-filename }|
            filetype = 'BIN'
          IMPORTING
            filelength = lv_filesize
          CHANGING
            data_tab = lt_bin_data
          EXCEPTIONS
            file_open_error         = 1
            file_read_error         = 2
            no_batch                = 3
            gui_refuse_filetransfer = 4
            invalid_type            = 5
            no_authority            = 6
            unknown_error           = 7
            bad_data_format         = 8
            header_not_allowed      = 9
            separator_not_allowed   = 10
            header_too_long         = 11
            unknown_dp_error        = 12
            access_denied           = 13
            dp_out_of_memory        = 14
            disk_full               = 15
            dp_timeout              = 16
            not_supported_by_gui    = 17
            error_no_gui            = 18
            OTHERS                  = 19
        ).
        IF sy-subrc <> 0.
          LEAVE PROGRAM.
        ENDIF.
"START OF ADD BY JIMBED
try. "NEW CODE
        lt_nbin_data = lt_bin_data.
        lv_nfilesize = lv_filesize.
        "this is for solix-
>xstring (convert data from the table into a long single string)
        DATA(gv_bin_data) = cl_bcs_convert=>solix_to_xstring( it_solix = lt_b
in_data ).

        "this is to create spreadsheet ref object.
        DATA(lo_excel) = NEW cl_fdt_xl_spreadsheet( document_name = CONV #( l
t_files[ 1 ]-filename )
                                                       xdocument = gv_bin_dat
a ).
        lo_excel->if_fdt_doc_spreadsheet~get_worksheet_names(
          IMPORTING
            worksheet_names = lt_worksheet_names
        ).

"NEW CODE
```

```
catch cx_fdt_excel_core.
MESSAGE: TEXT-020 TYPE GC_S DISPLAY LIKE GC_E.
endtry.
"END OF ADD BY JIMBED

        IF lines( lt_worksheet_names ) > 0.
          "First worksheet ref to itab.
          DATA(lo_worksheet_ltab) = lo_excel-
>if_fdt_doc_spreadsheet~get_itab_from_worksheet( lt_worksheet_names[ 1 ] ).
          "this is ref to generic fieldsymbol (excel data).
          ASSIGN lo_worksheet_ltab->* TO FIELD-SYMBOL(<fs_worksheet0>).
          FREE lo_worksheet_ltab.

          "Seccond worksheet ref to itab.
          lo_worksheet_ltab = lo_excel-
>if_fdt_doc_spreadsheet~get_itab_from_worksheet( lt_worksheet_names[ 2 ] ).
          "this is ref to generic fieldsymbol2 (excel data).
          ASSIGN lo_worksheet_ltab->* TO FIELD-SYMBOL(<fs_worksheet1>).

          IF lv_po_mark EQ abap_true. "FOR Purchase Order header and Item.
            lt_s_header = <fs_worksheet0>.
            lt_s_itm = <fs_worksheet1>.

            IF lt_s_header[] IS NOT INITIAL.
              DELETE lt_s_header INDEX 1.
              DELETE lt_s_header WHERE keylink EQ space.
              LOOP AT lt_s_header INTO DATA(ls_s_header).
"START OF ADD BY JIMBED
                SHIFT ls_s_header-
vendor LEFT DELETING LEADING GC_0. "NEW CODE CONSTANT GC_0
"END OF ADD BY JIMBED
                TRANSLATE ls_s_header-doctyp TO UPPER CASE.
                ls_header-keylink   = ls_s_header-keylink.
                ls_header-doctyp   = ls_s_header-doctyp.
                ls_header-vendor    = |00{ ls_s_header-vendor }|.
                ls_header-purch_org = ls_s_header-purch_org.
                ls_header-purch_grp = ls_s_header-purch_grp.
                ls_header-comp_code = ls_s_header-comp_code.

                APPEND ls_header TO lt_header.
                FREE: ls_header, ls_s_header.
              ENDLOOP.
            ENDIF.

            IF lt_s_itm[] IS NOT INITIAL.
              DELETE lt_s_itm INDEX 1.
              LOOP AT lt_s_itm INTO DATA(ls_s_itm).
                TRANSLATE ls_s_itm-matnr TO UPPER CASE.
                ls_itm-keylink = ls_s_itm-keylink.
                ls_itm-matnr   = ls_s_itm-matnr.
                ls_itm-qty     = ls_s_itm-qty.
                ls_itm-price   = ls_s_itm-price.
                ls_itm-plant   = ls_s_itm-plant.

                APPEND ls_itm TO lt_itm.
                FREE: ls_itm, ls_s_itm.
              ENDLOOP.
            ENDIF.
          ELSE. "FOR Sales Order header and Item.
            lt_s_soheader = <fs_worksheet0>.
            lt_s_soitm = <fs_worksheet1>.

            IF lt_s_soheader[] IS NOT INITIAL.
              DELETE lt_s_soheader INDEX 1.
              DELETE lt_s_soheader WHERE keylink EQ space.
              LOOP AT lt_s_soheader INTO DATA(ls_s_soheader).
                TRANSLATE ls_s_soheader-doctyp TO UPPER CASE.
                TRANSLATE ls_s_soheader-custporef TO UPPER CASE.
                TRANSLATE ls_s_soheader-prtnr1 TO UPPER CASE.
                TRANSLATE ls_s_soheader-prtnr2 TO UPPER CASE.

                CALL FUNCTION 'CONVERSION_EXIT_AUART_INPUT'
                  EXPORTING
                    input  = ls_s_soheader-doctyp
                  IMPORTING
                    output = ls_s_soheader-doctyp.

                ls_soheader-keylink   = ls_s_soheader-keylink.
                ls_soheader-doctyp    = ls_s_soheader-doctyp.
                ls_soheader-salesorg  = ls_s_soheader-salesorg.
                ls_soheader-dchannel  = ls_s_soheader-dchannel.
                ls_soheader-division  = ls_s_soheader-division.
                ls_soheader-custporef = ls_s_soheader-custporef.
                ls_soheader-prtnr1    = ls_s_soheader-prtnr1.
                ls_soheader-prtnr2    = ls_s_soheader-prtnr2.

                APPEND ls_soheader TO lt_soheader.
                FREE: ls_soheader, ls_s_soheader.
              ENDLOOP.
            ENDIF.

            IF lt_s_soitm[] IS NOT INITIAL.
              DELETE lt_s_soitm INDEX 1.
              LOOP AT lt_s_soitm INTO DATA(ls_s_soitm).
                TRANSLATE ls_s_soitm-matr TO UPPER CASE.
                ls_soitm-keylink = ls_s_soitm-keylink.
                ls_soitm-matr   = ls_s_soitm-matr.
                ls_soitm-qty     = ls_s_soitm-qty.
```

```
                APPEND ls_soitm TO lt_soitm.
                FREE: ls_soitm, ls_s_soitm.
              ENDLOOP.
            ENDIF.
          ENDIF.

          ELSE.
"START OF ADD BY JIMBED
MESSAGE: TEXT-020 TYPE GC_S DISPLAY LIKE GC_E.
"END OF ADD BY JIMBED
        ENDIF.

      ENDIF.
    ENDIF.


  ENDMETHOD.
```

Method Name: CREATE_SO_NO

Parameters:



Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Parameters of Method: CREATE_SO_NO

Parameter	Type	Pa...	Op...	Typing Method	Associated Type	Default Value	Description
LT_SO_ITM	Importing	☑	□	Type	TY_SOITM		SO Item Standard Table
LT_SOHEADER	Importing	☑	□	Type	TY_SOHEADER		SO Header Standard Table
LT_SORETURN	Exporting	☑	□	Type	TY_RETURN		PO Return Parameter
LV_SOVBELN	Exporting	☑	□	Type	BAPIVBELN-VBELN		Sales Document
LT_SO	Exporting	☑	□	Type	TY_SO		SO Number standard Table

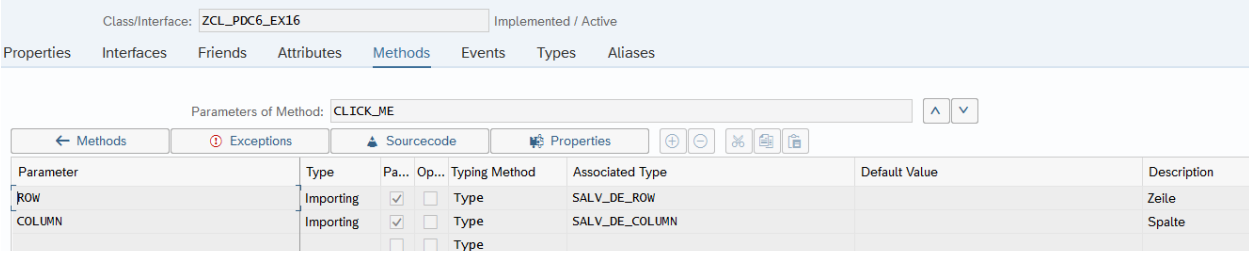
Source Code:

```
  METHOD create_so_no.
    DATA: lv_item                 TYPE posnr_va,
          ls_so                   TYPE ty_s,

          "BAPI DECLARATIONS.
          ls_order_header_in      TYPE bapisdhd1, "SO Header
          ls_order_header_inx     TYPE bapisdhd1x,

          lt_order_items_in       TYPE STANDARD TABLE OF bapisditm, "SO Items
          ls_order_items_in       TYPE bapisditm,

          lt_order_items_inx      TYPE STANDARD TABLE OF bapisditmx,
          ls_order_items_inx      TYPE bapisditmx,

          lt_order_partners       TYPE STANDARD TABLE OF bapiparnr, "SO Partne
```

```
rs
          ls_order_partners      TYPE bapiparnr,

          lt_order_schedules_in  TYPE STANDARD TABLE OF bapischdl, "SO Sched
Items
          ls_order_schedules_in  TYPE bapischdl,

          lt_order_schedules_inx TYPE STANDARD TABLE OF bapischdlx,
          ls_order_schedules_inx TYPE bapischdlx.

    CONSTANTS: lc_mark_i TYPE char1 VALUE 'I',
               lc_mark_ag TYPE char2 VALUE 'AG',
               lc_mark_re TYPE char2 VALUE 'RE'.

    LOOP AT lt_soheader INTO DATA(ls_soheader) WHERE keylink NE space.

      ls_order_header_in-doc_type    = ls_soheader-doctyp.
      ls_order_header_in-sales_org   = ls_soheader-salesorg.
      ls_order_header_in-distr_chan  = ls_soheader-dchannel.
      ls_order_header_in-division    = ls_soheader-division.
      ls_order_header_in-purch_no_c  = ls_soheader-custporef.

      ls_order_header_inx-updateflag   = lc_mark_i.
      ls_order_header_inx-doc_type     = abap_true.
      ls_order_header_inx-sales_org    = abap_true.
      ls_order_header_inx-distr_chan   = abap_true.
      ls_order_header_inx-division     = abap_true.
      ls_order_header_inx-purch_no_c   = abap_true.

      ls_order_partners-partn_role = lc_mark_ag.
      ls_order_partners-partn_numb = ls_soheader-prtnr1.
      APPEND ls_order_partners TO lt_order_partners.
      FREE: ls_order_partners.

      ls_order_partners-partn_role = lc_mark_re.
      ls_order_partners-partn_numb = ls_soheader-prtnr2.
      APPEND ls_order_partners TO lt_order_partners.
      FREE: ls_order_partners.

      LOOP AT lt_so_itm INTO DATA(ls_so_itm) WHERE keylink EQ ls_soheader-
keylink AND matr NE space.

        lv_item = lv_item + 1.

        ls_order_items_in-itm_number = lv_item.
        ls_order_items_in-material = ls_so_itm-matr.
        ls_order_items_in-target_qty  = ls_so_itm-qty.

        APPEND: ls_order_items_in TO lt_order_items_in.
        FREE:  ls_order_items_in.

        ls_order_items_inx-itm_number = lv_item.
        ls_order_items_inx-material    = abap_true.
        ls_order_items_inx-target_qty  = abap_true.
        APPEND: ls_order_items_inx TO lt_order_items_inx.
        FREE:  ls_order_items_inx.

        ls_order_schedules_in-itm_number = lv_item.
        ls_order_schedules_in-req_qty = ls_so_itm-qty.
        APPEND ls_order_schedules_in TO lt_order_schedules_in.
        FREE ls_order_schedules_in.

        ls_order_schedules_inx-itm_number = lv_item.
        ls_order_schedules_inx-req_qty = abap_true.
        APPEND ls_order_schedules_inx TO lt_order_schedules_inx.
        FREE ls_order_schedules_inx.
      ENDLOOP.

      CALL FUNCTION 'BAPI_SALESORDER_CREATEFROMDAT2'
        EXPORTING
          order_header_in     = ls_order_header_in
          order_header_inx    = ls_order_header_inx
        IMPORTING
          salesdocument       = lv_sovbeln
        TABLES
          return              = lt_soreturn
          order_items_in      = lt_order_items_in
          order_items_inx     = lt_order_items_inx
          order_partners      = lt_order_partners
          order_schedules_in  = lt_order_schedules_in
          order_schedules_inx = lt_order_schedules_inx.

      IF lv_sovbeln IS NOT INITIAL.
        CALL FUNCTION 'BAPI_TRANSACTION_COMMIT'
          EXPORTING
            wait = abap_true.
        CALL FUNCTION 'CONVERSION_EXIT_ALPHA_OUTPUT'
          EXPORTING
            input  = lv_sovbeln
          IMPORTING
            output = lv_sovbeln.
        IF sy-subrc EQ 0.
"START OF ADD BY JIMBED
SORT LT_SO BY KEYLINK ASCENDING.
"END OF ADD BY JIMBED
*          MESSAGE lv_sovbeln TYPE 'I'.
          ls_so-keylink = ls_soheader-keylink.
          ls_so-sono = lv_sovbeln.
          APPEND ls_so TO lt_so.
        ENDIF.
      ENDIF.

"START OF ADD BY JIMBED
if lt_soreturn is NOT INITIAL.
  CALL FUNCTION 'ZPDC7_ERRORS_MSG_SHOW'
    TABLES
      i_bapiret2_tab       = LT_SORETURN.
            .
  endif.
"END OF ADD BY JIMBED.

      FREE: ls_soheader, ls_so, ls_order_header_in, ls_order_header_inx, lt_o
rder_items_in, lt_soreturn,
      lt_order_items_inx, lt_order_partners, lt_order_schedules_in, lt_order_
schedules_inx.
    ENDLOOP.
  ENDMETHOD.
```

Method Name: CLICK_ME

Parameters:

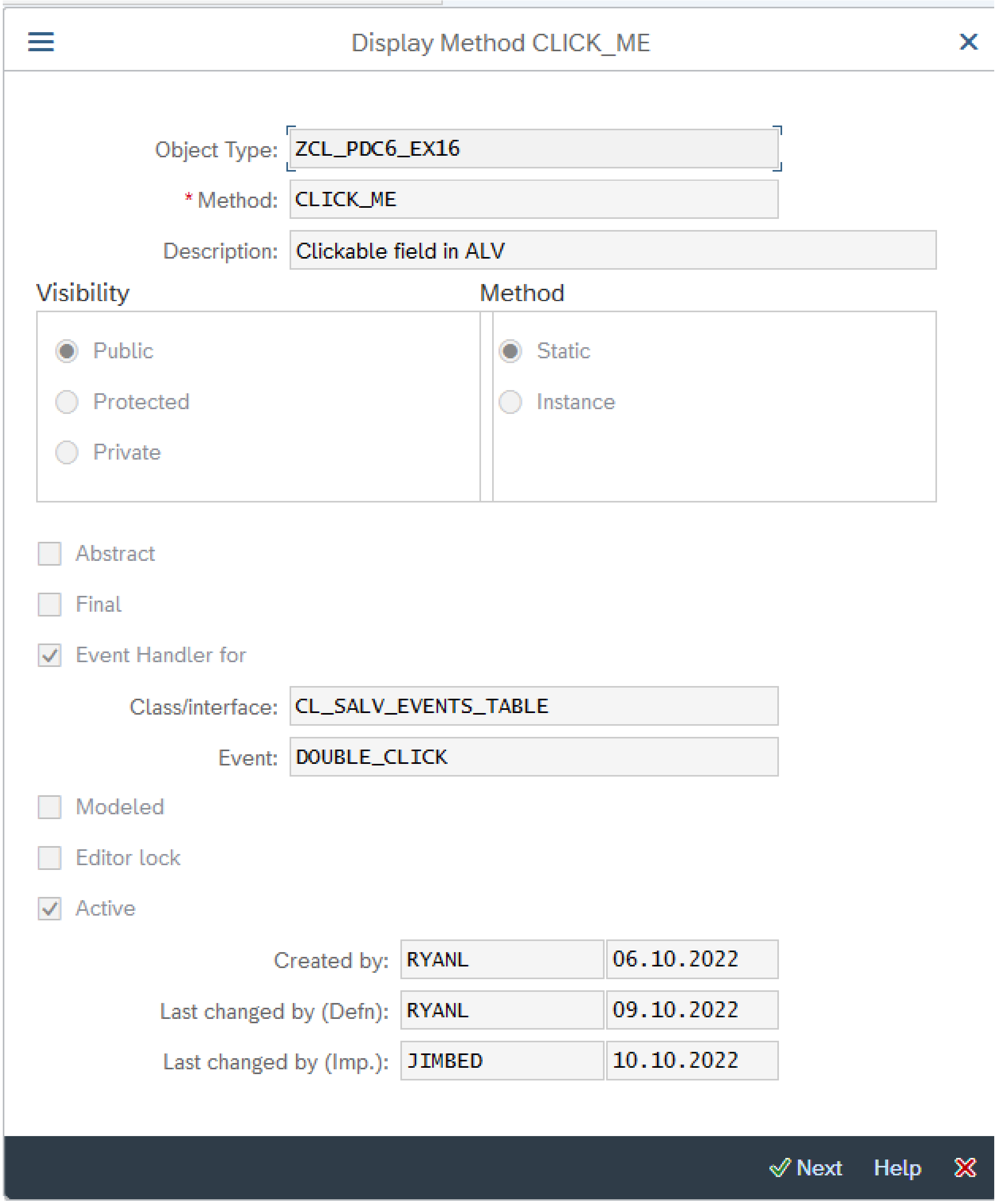


Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Parameters of Method: CLICK_ME

Parameter	Type	Pa...	Op...	Typing Method	Associated Type	Default Value	Description
ROW	Importing	☑	□	Type	SALV_DE_ROW		Zeile
COLUMN	Importing	☑	□	Type	SALV_DE_COLUMN		Spalte
		□	□	Type			

Properties:



Display Method CLICK_ME

Object Type:	ZCL_PDC6_EX16
* Method:	CLICK_ME
Description:	Clickable field in ALV

Visibility

● Public
○ Protected
○ Private

Method

● Static
○ Instance

□ Abstract
□ Final
☑ Event Handler for

Class/interface:	CL_SALV_EVENTS_TABLE
Event:	DOUBLE_CLICK

□ Modeled
□ Editor lock
☑ Active

Created by:	RYANL	06.10.2022
Last changed by (Defn):	RYANL	09.10.2022
Last changed by (Imp.):	JIMBED	10.10.2022

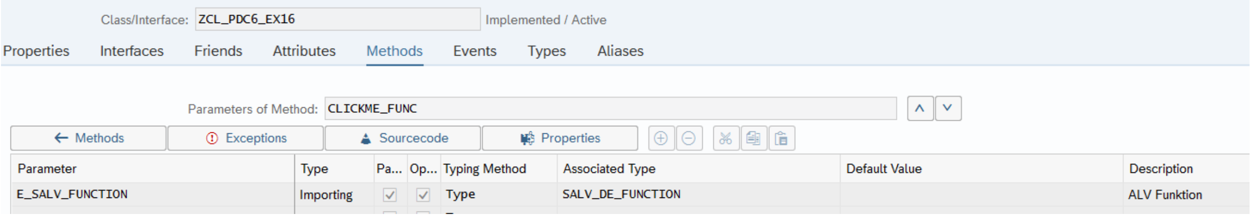
Source Code:

```
METHOD click_me.
  DATA lv_postring TYPE char40.
  IF lv_so_mark EQ abap_true.
```

```
      IF column EQ TEXT-014.
        READ TABLE lt_soalv INTO DATA(ls_soalv) INDEX row.

        CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
          EXPORTING
            input  = ls_soalv-sono
          IMPORTING
            output = ls_soalv-sono.

        SET PARAMETER ID text-015 FIELD ls_soalv-sono.
        CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.
      ENDIF.
    ELSE.
      IF column EQ text-016.
        READ TABLE lt_poalv INTO DATA(ls_poalv) INDEX row.

        CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
          EXPORTING
            input  = ls_poalv-pono
          IMPORTING
            output = ls_poalv-pono.

        SET PARAMETER ID text-018 FIELD ls_poalv-pono.
        CALL TRANSACTION 'ME23N'.
      ENDIF.
    ENDIF.
  ENDMETHOD.
```

Method Name: CLICKME_FUNC

Parameters:

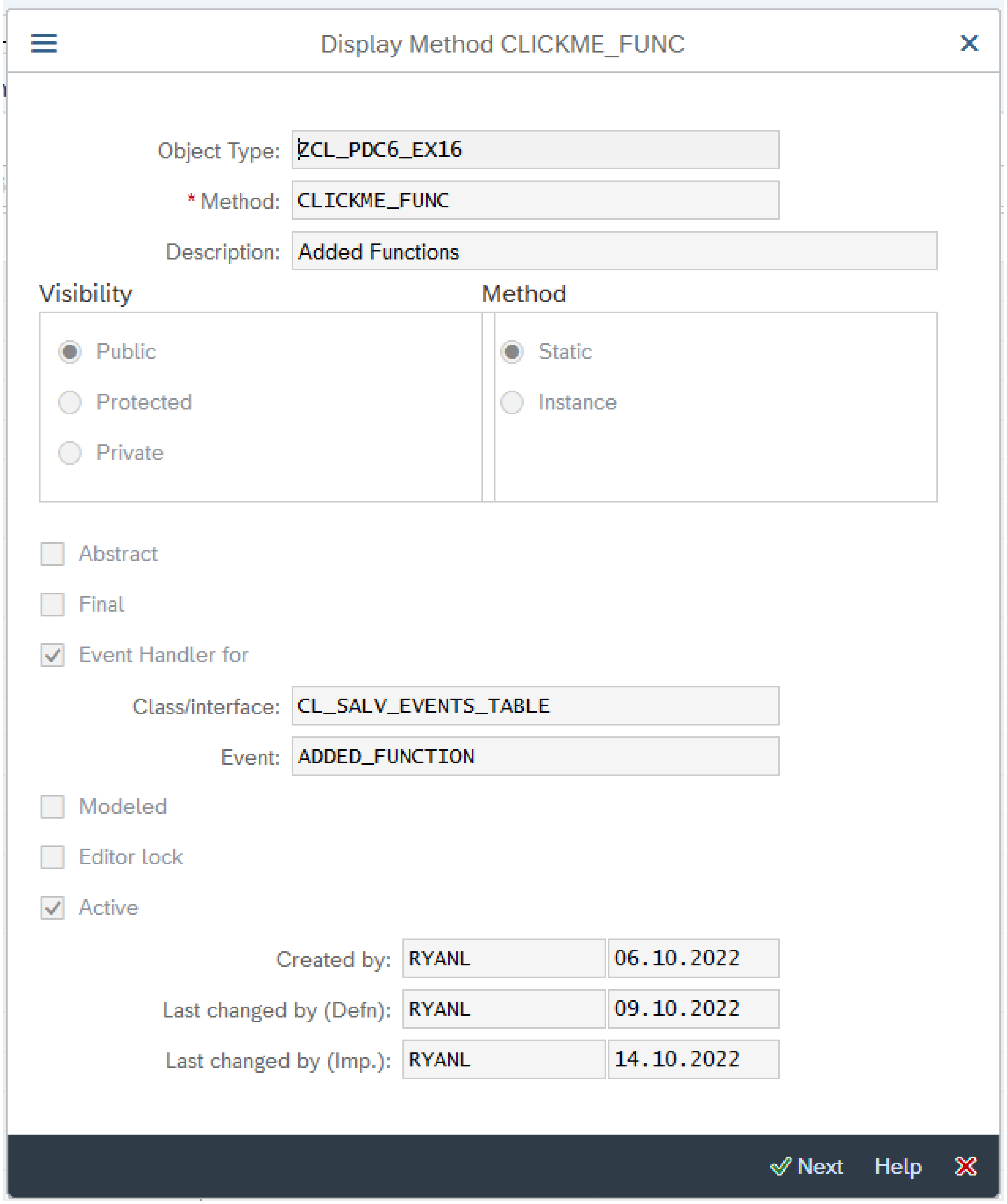


Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Parameters of Method: CLICKME_FUNC

Parameter	Type	Pa...	Op...	Typing Method	Associated Type	Default Value	Description
E_SALV_FUNCTION	Importing	☑	☑	Type	SALV_DE_FUNCTION		ALV Funktion

Properties:



Display Method CLICKME_FUNC

Object Type: ZCL_PDC6_EX16

* Method: CLICKME_FUNC

Description: Added Functions

Visibility

● Public

○ Protected

○ Private

Method

● Static

○ Instance

☐ Abstract

☐ Final

☑ Event Handler for

Class/interface: CL_SALV_EVENTS_TABLE

Event: ADDED_FUNCTION

☐ Modeled

☐ Editor lock

☑ Active

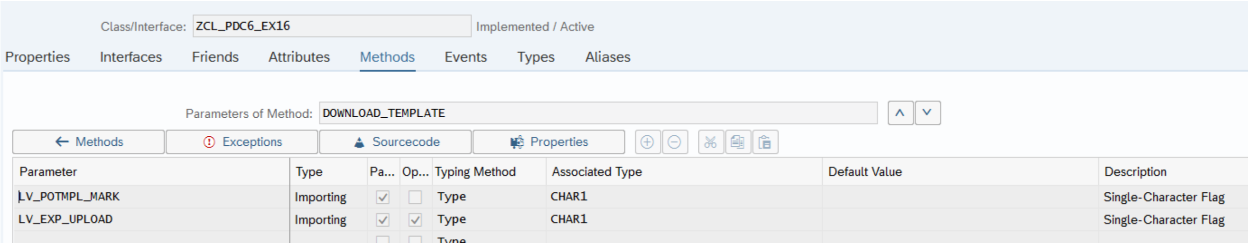
Created by:	RYANL	06.10.2022
Last changed by (Defn):	RYANL	09.10.2022
Last changed by (Imp.):	RYANL	14.10.2022

Source Code:

```
METHOD clickme_func.
  IF e_salv_function EQ '%ZSMAIL'.
    SUBMIT zabap_ex16mail_pdc6 VIA SELECTION-SCREEN AND RETURN.
  ENDIF.
ENDMETHOD.
```

Method Name: DOWNLOAD_TEMPLATE

Parameters:



Class/Interface: ZCL_PDC6_EX16 Implemented / Active

Parameters of Method: DOWNLOAD_TEMPLATE

Parameter	Type	Pa...	Op...	Typing Method	Associated Type	Default Value	Description
LV_POTMPL_MARK	Importing	☑	□	Type	CHAR1		Single-Character Flag
LV_EXP_UPLOAD	Importing	☑	☑	Type	CHAR1		Single-Character Flag
		□	□	Type			

Source Code:

```
  METHOD download_template.
    DATA: lo_application TYPE ole2_object,
          lo_workbook    TYPE ole2_object,
          lo_sheet       TYPE ole2_object,
          lo_cells       TYPE ole2_object,

          lt_tblhead     TYPE STANDARD TABLE OF zcl_pdc6_ex16=>ty_hstring, "P
O header.
          lt_tblitem     TYPE STANDARD TABLE OF zcl_pdc6_ex16=>ty_istring, "P
O Item.
          lt_tblsohead   TYPE STANDARD TABLE OF zcl_pdc6_ex16=>ty_soh, "SO he
ader.
          lt_tblsoitem   TYPE STANDARD TABLE OF zcl_pdc6_ex16=>ty_soi, "SO It
em.
          lt_colheader   TYPE STANDARD TABLE OF zcl_pdc6_ex16=>ty_col,
          lt_colitem     TYPE STANDARD TABLE OF zcl_pdc6_ex16=>ty_col,
          lo_itab        TYPE REF TO cl_abap_tabledescr,
          lo_struct      TYPE REF TO cl_abap_structdescr,
          lv_col         TYPE i,
          lv_filename    TYPE string
          .
    CONSTANTS: gc_row TYPE i VALUE 1,
               gc_col TYPE i VALUE 0.

    FIELD-SYMBOLS: <fs_table>  TYPE ANY TABLE, "FS_TABLE FOR HEADER
                   <fs_table1> TYPE ANY TABLE. "FS_TABLE FOR ITEMS

* TABLE HEADER FOR SO AND PO.
    IF lv_potmpl_mark EQ abap_true.
      lv_filename = 'c:\Purchase Order.xlsx'.
      ASSIGN lt_tblhead TO <fs_table>.
      ASSIGN lt_tblitem TO <fs_table1>.
    ELSE.
```

```
      lv_filename = 'c:\Sales Order.xlsx'.
      ASSIGN lt_tblsohead TO <fs_table>.
      ASSIGN lt_tblsoitem TO <fs_table1>.
    ENDIF.

    IF lv_exp_upload EQ space.
      lo_itab ?= cl_abap_typedescr=>describe_by_data( <fs_table> ).
      DATA(lo_line) = lo_itab->get_table_line_type( ).

      IF lo_line->kind = cl_abap_typedescr=>kind_struct.
        lo_struct ?= lo_line.
        LOOP AT lo_struct->components[] ASSIGNING FIELD-SYMBOL(<fs_comp>).
          DATA(lv_fieldname) = VALUE zcl_pdc6_ex16=>ty_col( colname = <fs_com
p>-name ).
          APPEND lv_fieldname TO lt_colheader.
          FREE lv_fieldname.
        ENDLOOP.
      ENDIF.

      FREE: lo_itab, lo_line, lo_struct.

* TABLE ITEM FOR SO AND PO.
*      IF lv_potmpl_mark EQ abap_true.
*        ASSIGN lt_tblitem TO <fs_table1>.
*      ELSE.
*        ASSIGN lt_tblsoitem TO <fs_table1>.
*      ENDIF.

      lo_itab ?= cl_abap_typedescr=>describe_by_data( <fs_table1> ).
      lo_line = lo_itab->get_table_line_type( ).

      IF lo_line->kind = cl_abap_typedescr=>kind_struct.
        lo_struct ?= lo_line.
        LOOP AT lo_struct->components[] ASSIGNING FIELD-SYMBOL(<fs_comp1>).
          lv_fieldname = VALUE zcl_pdc6_ex16=>ty_col( colname = <fs_comp1>-
name ).
          APPEND lv_fieldname TO lt_colitem.
          FREE lv_fieldname.
        ENDLOOP.
      ENDIF.
    ENDIF.


    IF lt_colitem[] IS NOT INITIAL.
      lv_col = 0.
      CREATE OBJECT lo_application TEXT-001.
      SET PROPERTY OF lo_application TEXT-002 = 1.
      CALL METHOD OF lo_application TEXT-
003 = lo_workbook. "this is to create empty workbooks.
```

```
* This is for second sheet
      CALL METHOD OF lo_workbook TEXT-004 = lo_workbook.
      GET PROPERTY OF lo_application TEXT-005 = lo_sheet.
      SET PROPERTY OF lo_sheet TEXT-006 = TEXT-007.

      LOOP AT lt_colitem INTO DATA(gs_colitem).
        lv_col = lv_col + gc_row.
        CALL METHOD OF lo_sheet TEXT-008 = lo_cells
          EXPORTING
            #1 = gc_row "row
            #2 = lv_col. "col

        SET PROPERTY OF lo_cells TEXT-009 = gs_colitem-colname.
        FREE gs_colitem.
      ENDLOOP.
    ENDIF.

    IF lt_colheader[] IS NOT INITIAL.
      lv_col = gc_col.
* This is for first sheet.
      GET PROPERTY OF lo_application TEXT-010 = lo_sheet.
      CALL METHOD OF lo_sheet TEXT-004 = lo_workbook.

      GET PROPERTY OF lo_application TEXT-012 = lo_sheet.
      SET PROPERTY OF lo_sheet TEXT-006 = TEXT-011.

      LOOP AT lt_colheader INTO DATA(gs_colheader).
        lv_col = lv_col + gc_row.
        CALL METHOD OF lo_sheet TEXT-008 = lo_cells
          EXPORTING
            #1 = gc_row "row
            #2 = lv_col. "col

        SET PROPERTY OF lo_cells TEXT-009 = gs_colheader-colname.
        FREE gs_colheader.
      ENDLOOP.
    ENDIF.

*SAVE EXCEL SPREADSHEET TO PARTICULAR FILENAME
    CALL METHOD OF lo_sheet TEXT-013
      EXPORTING
        #1 = lv_filename. "filename

    FREE: lo_application, lo_workbook, lo_sheet, lo_cells,
          lt_tblhead, lt_tblitem, lt_colheader, lt_colitem, lo_itab, lo_struc
t,
          lv_col, lv_filename.
```

```
  UNASSIGN: <fs_table>, <fs_table1>.
ENDMETHOD.
```