

```
*&-----*
*& Report   ZALV_TABLESTCODE                               *
*& Author   Swarna.S.                                     *
*&-----*
*& AS : ALV report to display the dictionary objects
*& (tables/structures/views of all types of delivery classes)
*& used by a TRANSACTION.The ALV is made interactive that on clicking
*& the dictionary object it goes to SE11 along with the object name in
*& memory
*&-----*
REPORT z_tablestcode .

*ALV type pools declarations

TYPE-POOLS : slis.

*Internal table and work area declarations for dd021 / dd02t /tstc

DATA : it_tstc TYPE STANDARD TABLE OF tstc,
      wa_tstc TYPE tstc,
      it_dd021 TYPE STANDARD TABLE OF dd021,
      wa_dd021 TYPE dd021,
      it_dd02t TYPE STANDARD TABLE OF dd02t,
      wa_dd02t TYPE dd02t.

*DATA DECLARATIONS FOR PROGRAM NAMES

DATA : progame LIKE sy-repid.

*DATA DECLARATIONS FOR tcode text

DATA : texts LIKE tstct-ttext.

*Structure for output

TYPES : BEGIN OF ty_output,
      tabname LIKE dd021-tabname,
      tabclass(20) TYPE c,
      contflag(80) TYPE c,
      text LIKE dd02t-ddtext,
      END OF ty_output.

*Internal table and work area declarations for output

DATA : it_output TYPE STANDARD TABLE OF ty_output,
      wa_output TYPE ty_output.

*Structure for table names

TYPES : BEGIN OF ty_names,
      name LIKE dd021-tabname,
      END OF ty_names.

*Internal table and work area declarations for table names

DATA : it_names TYPE STANDARD TABLE OF ty_names.

*data declarations for ALV
DATA: it_layout TYPE slis_layout_alv,
      wa_fieldcat TYPE slis_fieldcat_alv,
      it_fieldcat TYPE slis_t_fieldcat_alv.
```

***** SELECTION SCREEN *****

PARAMETERS : tcode LIKE sy-tcode.

***** INITIALIZATION *****
INITIALIZATION.

***** START OF SELECTION *****

START-OF-SELECTION.

*Select to check if the tcode exists

```
SELECT * FROM tstc INTO CORRESPONDING FIELDS OF TABLE it_tstc
WHERE tcode = tcode.
```

*If tcode does not exist message is thrown

```
IF sy-subrc <> 0.
```

```
  MESSAGE 'Enter a valid Transaction Code' TYPE 'I'.
```

```
  EXIT.
```

```
ENDIF.
```

*Assigning the program name form the tcode

```
READ TABLE it_tstc INTO wa_tstc WITH KEY tcode = tcode.
```

```
programe = wa_tstc-pgmna.
```

*to fetch the tcode texts

```
SELECT SINGLE ttext FROM tstct INTO texts WHERE tcode = tcode
AND sprsl = 'E'.
```

*to fetch the tables used

```
CALL FUNCTION 'GET_TABLES'
  EXPORTING
    programe = programe
  TABLES
    tables_tab = it_names.
```

*Check if there are tables in the internal tabel

```
IF it_names IS INITIAL.
```

```
  MESSAGE 'No Data Found' TYPE 'I'.
  EXIT.
```

```
ELSE.
```

*Subroutine to get the table details

```
  PERFORM tables_in_transaction.
```

```
ENDIF.
```

*output

```
PERFORM alv_output.

*&-----*
*& Form TABLES_IN_TRANSACTION
*&-----*
* text
*-----*

FORM tables_in_transaction .

*To fetch Tables and their features

  IF it_names[] IS NOT INITIAL.

    SELECT tabname tabclass contflag FROM dd02l
    INTO CORRESPONDING FIELDS OF TABLE it_dd02l
    FOR ALL ENTRIES IN it_names
    WHERE tabname EQ it_names-name.

  ENDIF.

*To fetch the texts for the table
  IF it_dd02l[] IS NOT INITIAL.

    SELECT tabname ddtxt FROM dd02t INTO CORRESPONDING FIELDS OF TABLE
it_dd02t
    FOR ALL ENTRIES IN it_dd02l WHERE tabname EQ it_dd02l-tabname AND
ddlanguge = 'E'.

  ENDIF.

*If no data is selected throw message

  IF sy-subrc <> 0.

    MESSAGE 'NO DATA FOUND' TYPE 'I'.
    EXIT.

  ENDIF.

*Appending values to the output table

  LOOP AT it_dd02l INTO wa_dd02l.

    wa_output-tabname = wa_dd02l-tabname.
    wa_output-tabclass = wa_dd02l-tabclass.
    wa_output-contflag = wa_dd02l-contflag.

    READ TABLE it_dd02t INTO wa_dd02t WITH KEY tabname = wa_dd02l-tabname.

    wa_output-text = wa_dd02t-ddtxt.

    APPEND wa_output TO it_output.
    CLEAR wa_output.

  ENDLLOOP.

*modifying the values in the output table for texts

  LOOP AT it_output INTO wa_output.
```

```
AT NEW tablename.
  READ TABLE it_dd021 INTO wa_dd021 WITH KEY tablename = wa_output-tabname.

  CASE wa_dd021-contflag.
    WHEN 'A'.
      wa_output-contflag = 'Application table (master and transaction
data)'.
    WHEN 'C'.
      wa_output-contflag = 'Customizing table, maintenance only by cust.,
not SAP import '.
    WHEN 'L'.
      wa_output-contflag = 'Table for storing temporary data, delivered
empty'.
    WHEN 'G'.
      wa_output-contflag = 'Customizing table, protected against SAP Upd.,
only INS all'.
    WHEN 'E'.
      wa_output-contflag = 'Control table, SAP and customer have separate
key areas '.
    WHEN 'S'.
      wa_output-contflag = 'System table, maint. only by SAP, change =
modification'.
    WHEN 'W'.
      wa_output-contflag = 'System table, contents transportable via
separate TR objects '.
    WHEN ' '.
      wa_output-contflag = 'Delivery class not available '.

  ENDCASE.

  CASE wa_dd021-tabclass.
    WHEN 'TRANSP'.
      wa_output-tabclass = 'Transparent table'.
    WHEN 'INTTAB'.
      wa_output-tabclass = 'Structure'.
    WHEN 'CLUSTER'.
      wa_output-tabclass = 'Cluster table'.
    WHEN 'POOL'.
      wa_output-tabclass = 'Pooled table'.
    WHEN 'VIEW'.
      wa_output-tabclass = 'General view structure '.
    WHEN 'APPEND'.
      wa_output-tabclass = 'Append structure'.

  ENDCASE.

  MODIFY it_output FROM wa_output TRANSPORTING contflag
                                         tabclass
                                         WHERE tablename EQ wa_output-
tablename.
  CLEAR : wa_output , wa_dd021.

  ENDAT.
  ENDLLOOP.

ENDFORM. " TABLES_IN_TRANSACTION
```

```
*&-----*
*&      Form  ALV_OUTPUT
*&-----*
*      text
```

```
*-----*
FORM alv_output.

*FIELDCATALOGUE
  PERFORM build_fieldcat.

*LAYOUT
  PERFORM build_layout.

*DISPLAY
  PERFORM alv_display.

ENDFORM.                "ALV_OUTPUT
```

```
*&-----*
*&      Form  build_fieldcat
*&-----*
*      text
*-----*
*Field catalogue
FORM build_fieldcat.
```

```
  CLEAR wa_fieldcat.
  wa_fieldcat-row_pos  = '1'.
  wa_fieldcat-col_pos  = '1'.
  wa_fieldcat-fieldname = 'TABNAME'.
  wa_fieldcat-tabname  = 'IT_OUTPUT'.
  wa_fieldcat-seltext_m = 'TABLENAME'.
  APPEND wa_fieldcat TO it_fieldcat.
```

```
  CLEAR wa_fieldcat.
  wa_fieldcat-row_pos  = '1'.
  wa_fieldcat-col_pos  = '2'.
  wa_fieldcat-fieldname = 'TABCLASS'.
  wa_fieldcat-tabname  = 'IT_OUTPUT'.
  wa_fieldcat-seltext_m = 'CATEGORY'.
  APPEND wa_fieldcat TO it_fieldcat.
```

```
  CLEAR wa_fieldcat.
  wa_fieldcat-row_pos  = '1'.
  wa_fieldcat-col_pos  = '3'.
  wa_fieldcat-fieldname = 'TEXT'.
  wa_fieldcat-tabname  = 'IT_OUTPUT'.
  wa_fieldcat-seltext_m = 'DESCRIPTION'.
  APPEND wa_fieldcat TO it_fieldcat.
```

```
  CLEAR wa_fieldcat.
  wa_fieldcat-row_pos  = '1'.
  wa_fieldcat-col_pos  = '4'.
  wa_fieldcat-fieldname = 'CONTFLAG'.
  wa_fieldcat-tabname  = 'IT_OUTPUT'.
  wa_fieldcat-seltext_m = 'Delivery Class'.
  APPEND wa_fieldcat TO it_fieldcat.
```

```
ENDFORM.                "build_fieldcat
```

```
*&-----*
*&      Form  build_layout
*&-----*
*      text
*-----*
*Layout
```

```
FORM build_layout.

  it_layout-zebra = 'X'.
  it_layout-colwidth_optimize = 'X'.

ENDFORM.                                "build_layout

*&-----*
*&      Form  alv_display
*&-----*
*      text
*-----*
*ALV output
FORM alv_display.

  CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
    EXPORTING
      i_callback_program           = sy-repid
      i_callback_html_top_of_page = 'HTML_TOP_OF_PAGE'
      i_callback_user_command      = 'USER_COMMAND'
      it_fieldcat                  = it_fieldcat
      is_layout                    = it_layout
    TABLES
      t_outtab                     = it_output.

ENDFORM.                                "alv_display

*-----*
*      FORM html_top_of_page
*-----*
FORM html_top_of_page USING top TYPE REF TO cl_dd_document.

  DATA tstring TYPE sdydo_text_element.
  DATA ttextss TYPE sdydo_text_element.

  tstring = tcode.

  ttextss = texts.

  CALL METHOD top->add_text
    EXPORTING
      text      = 'Tables used in the transaction'
      sap_style = 'Heading'.
  CALL METHOD top->add_text
    EXPORTING
      text      = tstring
      sap_style = 'Heading'.
  CALL METHOD top->add_text
    EXPORTING
      text      = ttextss
      sap_style = 'Heading'.

ENDFORM.                                "HTML_TOP_OF_PAGE

*&-----*
*&      Form  user_command
*&-----*
```

```
*      text
*-----*
*User actions on ALV
FORM user_command USING r_ucomm TYPE sy-ucomm
                    rs_selfield TYPE slis_selfield.

    CASE r_ucomm.

*User clicks a table name and control goes to sell with tablename from ALV
    WHEN '&IC1'.
        READ TABLE it_output INDEX rs_selfield-tabindex INTO wa_output.
        IF sy-subrc = 0.
            SET PARAMETER ID 'DTB' FIELD wa_output-tabname.
            CALL TRANSACTION 'SE11'.

        ENDIF.

    ENDCASE.

ENDFORM.                    "user_command
```

Output

Tables used in the transaction ME13 Display Purchasing Info Record

TABLERNAME	CATEGORY	DESCRIPTION	Delivery Class
A017	Pooled table	Material Info Record (Plant-Specific)	Application table (master and transaction data)
A018	Pooled table	Material Info Record	Application table (master and transaction data)

On clicking the table name it takes to SE11 screen as shown below.

The screenshot shows the SAP SE11 screen for table selection. The 'Database table' radio button is selected, and the value 'A017' is entered in the text field. Other options like 'View', 'Data type', 'Type Group', 'Domain', 'Search help', and 'Lock object' are unselected. At the bottom, there are three buttons: 'Display', 'Change', and 'Create'.