

```
*&-----*
*& Report   ZALV_TABLESPROG                               *
*& Author   Swarna.S.                                     *
*
*&-----*
*& AS : ALV report to display the dictionary objects
*& (tables/structures/views of all types of delivery classes)
*& used by a program.                                     *
*&                                                     *
*&-----*
```

REPORT ZALV_TABLESPROG .

*ALV type pools declarations
TYPE-POOLS : slis.

*Internal table and work area declarations for dd02l and dd02t
DATA : it_dd02l TYPE STANDARD TABLE OF dd02l,
 wa_dd02l TYPE dd02l,
 it_dd02t TYPE STANDARD TABLE OF dd02t,
 wa_dd02t TYPE dd02t.

*DATA DECLARATIONS FOR PROGRAM NAMES
DATA : progname LIKE sy-repid.

data : prognames(60) type c.

*Structure for output
TYPES : BEGIN OF ty_output,
 tabname LIKE dd02l-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 dd02l-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 : program LIKE sy-repid.

***** INITIALIZATION *****

INITIALIZATION.

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

START-OF-SELECTION.

*Select to check if the program exists
select single name from trdir into progname where name = program.

*If Program does not exist message is thrown
IF sy-subrc <> 0.

MESSAGE 'PROGRAM DOES NOT EXIST' TYPE 'I'.

EXIT.

ENDIF.

*Calling FM to get the tables associated with the program
progname = program.

CALL FUNCTION 'GET_TABLES'
EXPORTING
progname = progname
TABLES
tables_tab = it_names.

*Check if there are tables in the internal tabel
IF it_names IS INITIAL.

MESSAGE 'DATA DOES NOT EXIST' TYPE 'I'.

EXIT.

ELSE.

*Subroutine to get the table details
PERFORM TABLES_IN_PROGRAM.

ENDIF.

*output display
PERFORM alv_output.

```
*&-----*  
*& Form TABLES_IN_PROGRAM  
*&-----*  
* text  
*-----*
```

FORM TABLES_IN_PROGRAM.

*To fetch Tables and their features
IF it_names[] IS NOT INITIAL.

SELECT tabname tabclass contflag FROM dd021
INTO CORRESPONDING FIELDS OF TABLE it_dd021
FOR ALL ENTRIES IN it_names
WHERE tabname EQ it_names-name.

ENDIF.

*To fetch the texts for the table

```
IF it_dd021[] IS NOT INITIAL.

    SELECT tabname ddtex FROM dd02t INTO CORRESPONDING FIELDS OF TABLE
it_dd02t
    FOR ALL ENTRIES IN it_dd021 WHERE tabname EQ it_dd021-tabname AND
ddlanguag = 'E'.

ENDIF.

*If no data is selected throw message
IF sy-subrc <> 0.

    MESSAGE 'DATA DOES NOT EXIST' TYPE 'I'.

    EXIT.

ENDIF.

*Appending values to the output table
LOOP AT it_dd021 INTO wa_dd021.

    wa_output-tabname = wa_dd021-tabname.
    wa_output-tabclass = wa_dd021-tabclass.
    wa_output-contflag = wa_dd021-contflag.

    READ TABLE it_dd02t INTO wa_dd02t WITH KEY tabname = wa_dd021-tabname.

    wa_output-text = wa_dd02t-ddttext.

    APPEND wa_output TO it_output.
    CLEAR wa_output.

ENDLOOP.

*modifying the values in the output table for texts
LOOP AT it_output INTO wa_output.

    AT NEW tabname.
        READ TABLE it_dd021 INTO wa_dd021 WITH KEY tabname = 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 tabname EQ wa_output-
tabname.
    CLEAR : wa_output , wa_dd021.

    ENDAT.
    ENDLOOP.

ENDFORM. " TABLES_IN_PROGRAM

*&-----*
*&      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 = 'TABLENAM'.
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'
      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.

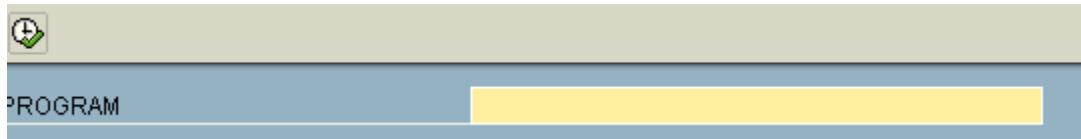
  tstring = program.

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

ENDFORM.

```

Selection screen



Output

Tables used in the program Z_CHECK			
TABlename	CATEGORY	DESCRIPTION	Delivery Class
ARC_PARAMS	Structure	ImageLink structure	Delivery class not available
BDCDATA	Structure	Batch input: New table field structure	Delivery class not available
BDCMSGCOLL	Structure	Collecting messages in the SAP System	Delivery class not available
CTU_PARAMS	Structure	Parameter string for runtime of CALL TRANSACTION USING...	Delivery class not available
PRI_PARAMS	Structure	Structure for Passing Print Parameters	Delivery class not available
RSJOBINFO	Structure	ABAP: Information about job scheduling	Delivery class not available
RSPARINT	Structure	PARAMETERS: Accompanying structure	Delivery class not available
RSSCR	Structure	Structure of internal table %_SSCR (selection screen, ABAP/4)	Delivery class not available
SCREEN	Structure	Structure Description for the System Data Object SCREEN	Delivery class not available
SSCRFIELDS	Structure	Fields on selection screens	Delivery class not available
SSCRTEXTS	Structure	Text fields on selection screens	Delivery class not available
SYST	Structure	ABAP System Fields	Delivery class not available
T100	Transparent table	Messages	Control table, SAP and customer have separate key areas
VARI	Transparent table	ABAP/4: Variant storage (similar to INDX)	System table, contents transportable via separate TR objects