```
REPORT zsf_find_text.

CLASS lcl_event_handeler DEFINITION DEFERRED.
DATA:lv_string TYPE tdsfname.

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.
SELECT-OPTIONS:s_form FOR lv_string.
PARAMETERS:p_forms  TYPE string NO-DISPLAY,
           find_dat TYPE string LOWER CASE OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1.

INITIALIZATION.
  s_form-low = 'Z*'.
  APPEND s_form.
  DATA i_formname TYPE string.
  DATA e_xml      TYPE xstring.
  TYPES:BEGIN OF ty_sf,
          formname TYPE tdsfname,
          formtype TYPE tdsftype,
          status   TYPE char8,
          e_xml    TYPE string,
        END OF ty_sf.
  DATA:lt_forms TYPE STANDARD TABLE OF ty_sf.

CLASS lcl_event_handeler DEFINITION.
  PUBLIC SECTION.
    METHODS: handle_hotspot FOR EVENT link_click OF cl_salv_events_table
      IMPORTING
          row
          column  .
ENDCLASS.
CLASS lcl_event_handeler IMPLEMENTATION.
  METHOD handle_hotspot.
    READ TABLE lt_forms INTO DATA(ls_forms) INDEX row.
    IF sy-subrc = 0.
      IF column = 'E_XML'.
        DATA: gcl_xml TYPE REF TO cl_xml_document.
        CREATE OBJECT gcl_xml.

*Parses XML String to DOM
        CALL METHOD gcl_xml->parse_string
          EXPORTING
            stream = ls_forms-e_xml.
*Display XML
        CALL METHOD gcl_xml->display.
      ELSE.

        CALL FUNCTION 'FB_DISPLAY_FORM'
```

```
        EXPORTING
          i_formname          = ls_forms-formname
          i_formtype          = ls_forms-formtype
          i_with_dialog       = abap_false
        EXCEPTIONS
          no_name             = 1
          no_form             = 2
          no_access_permission = 3
          illegal_language    = 4
          illegal_formtype    = 5
          OTHERS              = 6.
      IF sy-subrc <> 0.
* Implement suitable error handling here
      ENDIF.
    ENDIF.
  ENDIF.
 ENDMETHOD.
ENDCLASS.

START-OF-SELECTION.

 SELECT formname formtype FROM stxfadm INTO TABLE lt_forms
 WHERE formname IN s_form .
 LOOP AT lt_forms ASSIGNING FIELD-SYMBOL(<ls_forms>).
  DATA(lv_tabix) = sy-tabix.
  p_forms = <ls_forms>-formname.
  CALL FUNCTION 'FB_CONVERT_FORM_TO_XML'
   EXPORTING
    i_formname      = p_forms
   IMPORTING
    e_xml           = e_xml
   EXCEPTIONS
    no_active_source = 3.
  IF sy-subrc <> 0.
   <ls_forms>-status = 'Inactive'.
   CONTINUE.
  ENDIF.

  DATA ostr_output_data TYPE xstring.
  DATA codepage         TYPE cpcodepage.
  DATA cstr_output_data TYPE string.

  CALL FUNCTION 'SSFH_XSTRINGUTF8_TO_STRING'
   EXPORTING
    ostr_output_data = e_xml
   IMPORTING
    cstr_output_data = <ls_forms>-e_xml
   EXCEPTIONS
```

```
      conversion_error = 1
      internal_error   = 2
      OTHERS           = 3.
  IF sy-subrc <> 0.
    CONTINUE.
  ENDIF.

  SEARCH <ls_forms>-e_xml FOR find_dat AND MARK.
  IF sy-subrc NE 0.
    DELETE lt_forms INDEX lv_tabix.
  ENDIF.
ENDLOOP.


DATA: o_alv TYPE REF TO cl_salv_table.
DATA: lx_msg TYPE REF TO cx_salv_msg.
DATA: lr_columns TYPE REF TO cl_salv_columns,
      lr_column  TYPE REF TO cl_salv_column_table,
      lr_colums  TYPE REF TO cl_salv_columns_table,
      lr_display TYPE REF TO cl_salv_display_settings,
      title      TYPE lvc_title.
.
TRY.
    cl_salv_table=>factory(
      IMPORTING
        r_salv_table = o_alv
      CHANGING
        t_table      = lt_forms ).
  CATCH cx_salv_msg INTO lx_msg.
ENDTRY.
lr_display = o_alv->get_display_settings( ).
title = 'Search Results for Text:'(001) && | " | && find_dat && | " |.
lr_display->set_list_header( title  ).
lr_display->set_striped_pattern( if_salv_c_bool_sap=>true ).
lr_columns = o_alv->get_columns( ).
lr_columns->set_optimize( abap_true ).
DATA(gr_functions) = o_alv->get_functions( ).
gr_functions->set_all( abap_true ).
lr_colums  = o_alv->get_columns( ).
lr_column ?= lr_colums->get_column( 'FORMNAME' ).
lr_column->set_cell_type( if_salv_c_cell_type=>hotspot ).
lr_column ?= lr_colums->get_column( 'STATUS' ).
lr_column->set_long_text( 'Status' ).
lr_column->set_short_text( 'Status' ).
lr_column->set_medium_text( 'Status' ).
lr_column ?= lr_colums->get_column( 'FORMTYPE' ).
lr_column->set_visible( if_salv_c_bool_sap=>false ).
lr_column ?= lr_colums->get_column( 'E_XML' ).
```

```
  lr_column->set_cell_type( if_salv_c_cell_type=>hotspot ).
  DATA:co_report TYPE REF TO lcl_event_handeler.
  DATA: lo_events TYPE REF TO cl_salv_events_table.
  CREATE OBJECT co_report.
*   all events
  lo_events = o_alv->get_event( ).
*
*   event handler
  SET HANDLER co_report->handle_hotspot FOR lo_events.
  o_alv->display( ).

*Text elements
*----------------------------------------------------------
* 001 Search Results for Text:


*Selection texts
*----------------------------------------------------------
* FIND_DAT        Find Text
* S_FORM D       .

--------------------------------------------------------------------------------
```