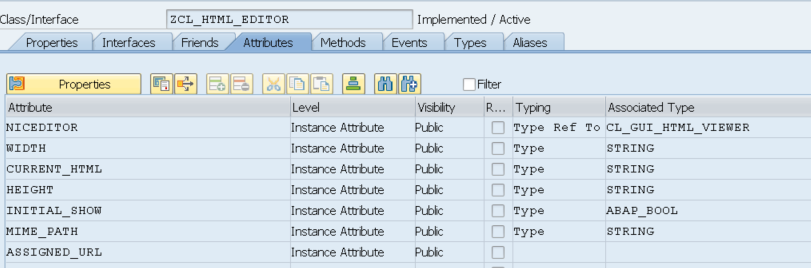
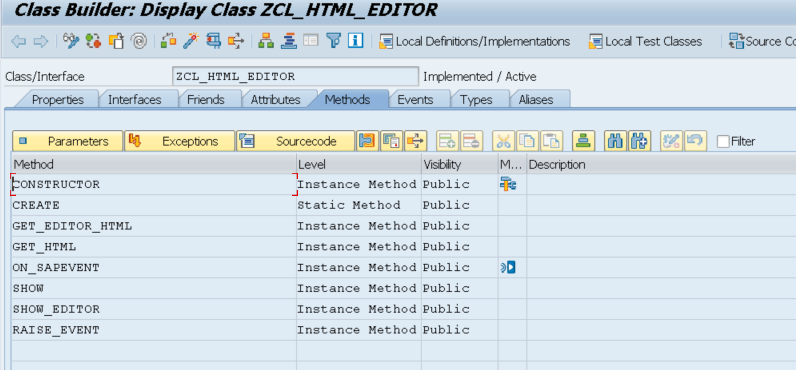
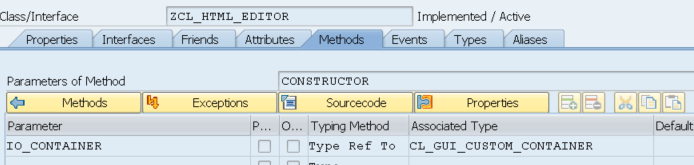
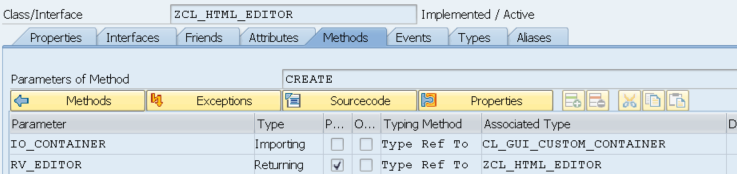
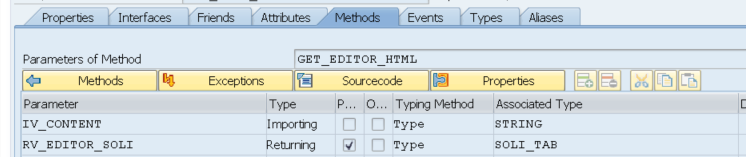
\*&---------------------------------------------------------------------\*  
\*& Report ZDEMO\_HTML\_EDITOR01  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zdemo\_html\_editor01.  
  
DATA: htmleditor TYPE REF TO zcl\_html\_editor.  
  
CLASS lcl\_event DEFINITION.  
  PUBLIC SECTION.  
    METHODS: html\_changed FOR EVENT html\_updated OF zcl\_html\_editor  
      IMPORTING html.  
  
ENDCLASS.  
  
  
DATA event\_handler TYPE REF TO lcl\_event.  
  
CLASS lcl\_event IMPLEMENTATION.  
  METHOD html\_changed.  
\*    cl\_demo\_output=>display\_html( html = html ).  
  ENDMETHOD.  
ENDCLASS.  
  
INITIALIZATION.  
  
  CALL SCREEN 0100.  
  
  
MODULE pbo OUTPUT.  
  SET PF-STATUS '0100'.  
  
  
  IF htmleditor IS INITIAL.  
    TRY .  
        htmleditor = zcl\_html\_editor=>create( io\_container = NEW cl\_gui\_custom\_container( container\_name = 'CC\_EDITOR' ) ).  
  
        htmleditor->show(  
          EXPORTING  
            iv\_html   = space  
            iv\_height = 600  
            iv\_width  = 1024 ).  
  
        CREATE OBJECT event\_handler.  
        SET HANDLER event\_handler->html\_changed FOR htmleditor.  
      CATCH cx\_root.  
  
    ENDTRY.  
  ENDIF.  
  
ENDMODULE.  
  
MODULE pai INPUT.  
  CASE sy-ucomm.  
    WHEN 'BACK' OR 'EXIT' OR 'LEAVE'.  
      LEAVE TO SCREEN 0.  
    WHEN 'SAVE'.  
      htmleditor->raise\_event( ).  
  ENDCASE.  
ENDMODULE.

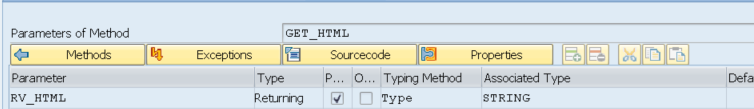


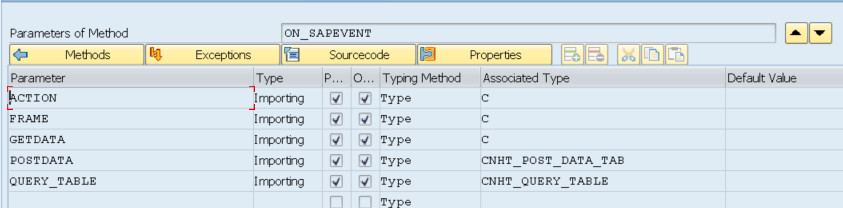


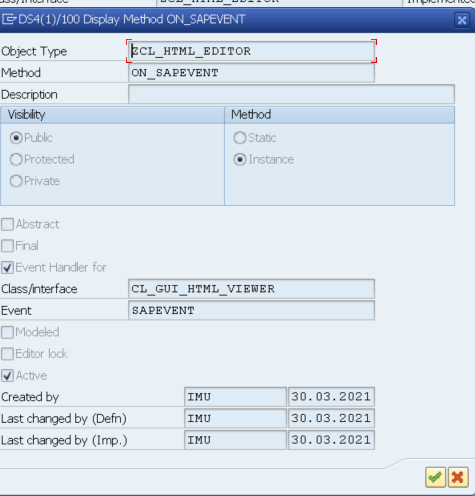


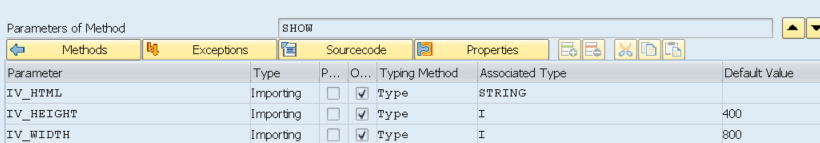


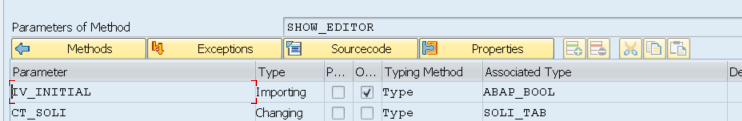


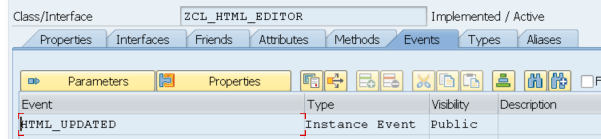


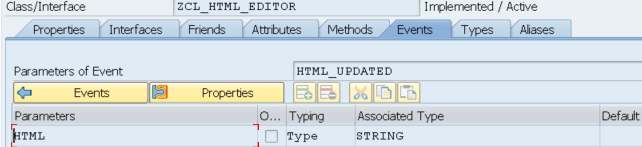












METHOD constructor.  
  
  CREATE OBJECT niceditor  
    EXPORTING  
      parent = io\_container.  
  
ENDMETHOD.

METHOD create.  
  
  CREATE OBJECT rv\_editor  
    EXPORTING  
      io\_container = io\_container.  
  
ENDMETHOD.

METHOD get\_editor\_html.  
  
  DATA: editor\_x TYPE xstring.  
  DATA: editor\_string TYPE string.  
  
  DATA: converter TYPE REF TO cl\_abap\_conv\_in\_ce.  
  
  cl\_mime\_repository\_api=>get\_api( )->get(  
                                           EXPORTING i\_url = mime\_path  
                                           IMPORTING e\_content = editor\_x  
                                           EXCEPTIONS OTHERS = 1 ).  
  CHECK sy-subrc EQ 0.  
  
  CALL METHOD cl\_abap\_conv\_in\_ce=>create  
    EXPORTING  
      encoding = 'UTF-8'  
      input    = editor\_x  
    RECEIVING  
      conv     = converter.  
  
  CALL METHOD converter->read  
    IMPORTING  
      data = editor\_string.  
  
  
  REPLACE ALL OCCURRENCES OF '&TextToReplace&' IN editor\_string WITH iv\_content.  
  REPLACE ALL OCCURRENCES OF '&&height&&'      IN editor\_string WITH height.  
  REPLACE ALL OCCURRENCES OF '&&width&&'       IN editor\_string WITH width.  
  
  rv\_editor\_soli =  cl\_bcs\_convert=>string\_to\_soli( iv\_string = editor\_string ).  
  
ENDMETHOD.

  METHOD get\_html.  
  
    rv\_html = current\_html.  
  ENDMETHOD.

METHOD on\_sapevent.  
  
  DATA: html TYPE string.  
  DATA: soli TYPE soli\_tab.  
  CHECK action EQ 'onSave'.  
  
  LOOP AT query\_table ASSIGNING FIELD-SYMBOL(<query>).  
    html = html && <query>-value.  
  ENDLOOP.  
  
  "update html  
  IF current\_html NE html.  
    current\_html = html.  
    RAISE EVENT html\_updated EXPORTING html = html.  
  ENDIF.  
  
  "as there was post method, we need to refresh  
  "html viewer with our html code  
  soli = get\_editor\_html( iv\_content = html ).  
  show\_editor( CHANGING ct\_soli = soli  ).  
  
ENDMETHOD.

METHOD show.  
  
  DATA: soli TYPE soli\_tab.  
  
  IF initial\_show EQ abap\_true OR iv\_height IS SUPPLIED AND iv\_width IS SUPPLIED.  
    height = iv\_height.  
    CONCATENATE height 'px' INTO height. CONDENSE height NO-GAPS.  
    width  = iv\_width.  
    CONCATENATE width 'px' INTO width. CONDENSE width NO-GAPS.  
  ENDIF.  
  "as there was post method, we need to refresh  
  "html viewer with our html code  
  current\_html = iv\_html.  
  soli = get\_editor\_html( iv\_content = iv\_html ).  
  show\_editor( EXPORTING iv\_initial = initial\_show CHANGING ct\_soli = soli  ).  
  
ENDMETHOD.

METHOD show\_editor.  
  
  DATA: events TYPE cntl\_simple\_events,  
        event  TYPE cntl\_simple\_event.  
  
  niceditor->load\_data(  
     EXPORTING  
       url                    = assigned\_url  
       type                   = 'text'    " Type of a MIME Object  
       subtype                = 'html'    " Subtype of a MIME Object  
     IMPORTING  
       assigned\_url           = assigned\_url  " URL  
     CHANGING  
       data\_table             = ct\_soli   " data table  
  
     EXCEPTIONS  
       dp\_invalid\_parameter   = 1  
       dp\_error\_general       = 2  
       cntl\_error             = 3  
       html\_syntax\_notcorrect = 4  
       OTHERS                 = 5 ).  
  IF sy-subrc >= 0.  
\*   message id sy-msgid type sy-msgty number sy-msgno  
\*              with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  IF iv\_initial EQ abap\_true.  
    initial\_show = abap\_false.  
\* register event  
    event-eventid = niceditor->m\_id\_sapevent.  
    event-appl\_event = 'x'.  
    APPEND event TO events.  
    CALL METHOD niceditor->set\_registered\_events  
      EXPORTING  
        events = events.  
  
    SET HANDLER on\_sapevent FOR niceditor.  
  
  ENDIF.  
  niceditor->show\_url( assigned\_url ).  
  
ENDMETHOD.

METHOD raise\_event.  
  
  RAISE EVENT html\_updated  
    EXPORTING  
      html   = current\_html  
    .  
  
ENDMETHOD.