  METHOD consultantidset\_create\_entity.  
    DATA : lt\_pdfcontent      TYPE TABLE OF solix,  
         lt\_binary\_content  TYPE solix\_tab,  
         lo\_bcs             TYPE REF TO cl\_bcs,  
         lo\_doc\_bcs         TYPE REF TO cl\_document\_bcs,  
         lv\_string\_text     TYPE string,  
         lt\_text            TYPE bcsy\_text,  
         lv\_sent\_to\_all     TYPE os\_boolean,  
         lo\_cx\_bcx          TYPE REF TO cx\_bcs,  
         lo\_recep           TYPE REF TO if\_recipient\_bcs,  
         lt\_lines           TYPE TABLE OF tline,  
         lv\_attachment\_text TYPE so\_obj\_des,  
         lv\_date            TYPE c LENGTH 10,  
         lv\_formatted\_date            TYPE c LENGTH 10,  
         lv\_bin\_filesize    TYPE so\_obj\_len,  
         ls\_mails           TYPE adr6-smtp\_addr,  
         lv\_mails           TYPE adr6-smtp\_addr,  
         lt\_mails           TYPE TABLE OF adr6-smtp\_addr.  
  
  DATA : lt\_consultant TYPE TABLE OF ztab\_gest\_cons,  
         ls\_consultant TYPE ztab\_gest\_cons.  
  
  DATA : lt\_manager TYPE TABLE OF ztab\_gest\_manage,  
         ls\_manager TYPE ztab\_gest\_manage,  
         lv\_email         TYPE adr6-smtp\_addr.  
  
  *"old*  
    DATA : ws\_entity LIKE er\_entity,  
           ls\_data   TYPE ztab\_gest\_cons.  
  
    io\_data\_provider->read\_entry\_data(  
   IMPORTING  
     es\_data = ws\_entity  
   ).  
  MOVE-CORRESPONDING ws\_entity TO ls\_data.  
  
    select single consultant\_id  
      from ztab\_gest\_cons  
      where consultant\_id eq @ls\_data-consultant\_id  
      into @data(ls\_cons).  
  
    MODIFY ztab\_gest\_cons FROM ls\_data.  
    MOVE-CORRESPONDING ws\_entity TO ls\_data.  
  *"old*  
  
  if ls\_cons is INITIAL.  
    TRY.  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
        *" Get email's body and put it in lt\_lines*  
        lo\_bcs = cl\_bcs=>create\_persistent( ).  
        CALL FUNCTION 'READ\_TEXT'  
          EXPORTING  
            id       = 'ST'  
            language = 'F'  
            name     = 'ZMAIL\_DISPON\_CREATE\_USER'  
            object   = 'TEXT'  
          TABLES  
            lines    = lt\_lines  
          EXCEPTIONS  
            id       = 1  
            language = 2  
            name     = 3  
            object   = 4  
            OTHERS   = 5.  
        IF sy-subrc <> 0.  
          MESSAGE 'Error reading text' TYPE 'E'.  
          RETURN.  
        ENDIF.  
  
  
  
        *" Dynamic values*  
        LOOP AT lt\_lines ASSIGNING FIELD-SYMBOL(<fs\_data1>).  
          REPLACE '&Login&'   WITH ls\_data-login  INTO <fs\_data1>-tdline.  
          REPLACE '&Password&'    WITH ls\_data-password      INTO <fs\_data1>-tdline.  
          REPLACE '&Role&'      WITH 'Consultant' INTO <fs\_data1>-tdline.  
          APPEND <fs\_data1>-tdline TO lt\_text.  
        ENDLOOP.  
  
        *" Create Document*  
        lo\_doc\_bcs = cl\_document\_bcs=>create\_document(  
                        i\_type    = 'RAW'  
                        i\_text    = lt\_text[]  
                        i\_length  = '12'  
                        i\_subject = 'Account creation.' ).  
  
        CALL METHOD lo\_bcs->set\_document( lo\_doc\_bcs ).  
  
  
        lo\_recep = cl\_cam\_address\_bcs=>create\_internet\_address( 'equipe.BC@inetum.ma' ).  
  
        CALL METHOD lo\_bcs->add\_recipient  
          EXPORTING  
            i\_recipient = lo\_recep  
            i\_express   = 'X'.  
  
        *" Send email immediately*  
        CALL METHOD lo\_bcs->set\_send\_immediately  
          EXPORTING  
            i\_send\_immediately = 'X'.  
  
        *" Send the email*  
        CALL METHOD lo\_bcs->send(  
          EXPORTING  
            i\_with\_error\_screen = 'X'  
          RECEIVING  
            result              = lv\_sent\_to\_all ).  
  
        *" Check if email was sent successfully*  
        IF lv\_sent\_to\_all IS INITIAL.  
          MESSAGE 'Email not sent' TYPE 'E'.  
        ELSE.  
          MESSAGE 'Email sent successfully' TYPE 'S'.  
          MODIFY ztab\_gest\_cons FROM ls\_data.  
        ENDIF.  
  
    CATCH cx\_bcs INTO lo\_cx\_bcx.  
      *" Log the exception*  
      WRITE: 'Exception:', lo\_cx\_bcx->error\_type.  
  ENDTRY.  
  ENDIF.  
  
  
ENDMETHOD.

  method CONSULTANTIDSET\_DELETE\_ENTITY.  
*\*\*TRY.*  
*\*CALL METHOD SUPER->CONSULTANTIDSET\_DELETE\_ENTITY*  
*\*  EXPORTING*  
*\*    IV\_ENTITY\_NAME          =*  
*\*    IV\_ENTITY\_SET\_NAME      =*  
*\*    IV\_SOURCE\_NAME          =*  
*\*    IT\_KEY\_TAB              =*  
*\*\*    io\_tech\_request\_context =*  
*\*    IT\_NAVIGATION\_PATH      =*  
*\*    .*  
*\*\*  CATCH /iwbep/cx\_mgw\_busi\_exception.*  
*\*\*  CATCH /iwbep/cx\_mgw\_tech\_exception.*  
*\*\*ENDTRY.*  
    DATA: lv\_id\_cons TYPE ztab\_gest\_cons-consultant\_id.  
    READ TABLE it\_key\_tab INTO DATA(ls\_key) WITH KEY name = 'ConsultantId' .  
    IF sy-subrc EQ 0.  
       MOVE ls\_key-value TO lv\_id\_cons.  
    ENDIF.  
    DELETE FROM ztab\_gest\_cons WHERE consultant\_id EQ lv\_id\_cons.  
  
  endmethod.

  method CONSULTANTIDSET\_GET\_ENTITY.  
*\*\*TRY.*  
*\*CALL METHOD SUPER->CONSULTANTIDSET\_GET\_ENTITY*  
*\*  EXPORTING*  
*\*    IV\_ENTITY\_NAME          =*  
*\*    IV\_ENTITY\_SET\_NAME      =*  
*\*    IV\_SOURCE\_NAME          =*  
*\*    IT\_KEY\_TAB              =*  
*\*\*    io\_request\_object       =*  
*\*\*    io\_tech\_request\_context =*  
*\*    IT\_NAVIGATION\_PATH      =*  
*\*\*  IMPORTING*  
*\*\*    er\_entity               =*  
*\*\*    es\_response\_context     =*  
*\*    .*  
*\*\*  CATCH /iwbep/cx\_mgw\_busi\_exception.*  
*\*\*  CATCH /iwbep/cx\_mgw\_tech\_exception.*  
*\*\*ENDTRY.*  
    DATA: lv\_id\_cons TYPE ztab\_gest\_cons-consultant\_id.  
  
    READ TABLE it\_key\_tab INTO DATA(ls\_data) WITH KEY name = 'ConsultantId' .  
    IF sy-subrc EQ 0.  
      lv\_id\_cons  = ls\_data-value.  
    ENDIF.  
  
    SELECT SINGLE \* FROM ztab\_gest\_cons INTO  er\_entity WHERE consultant\_id EQ lv\_id\_cons.  
  
  endmethod.

  METHOD consultantidset\_get\_entityset.  
*\*\*TRY.*  
*\*CALL METHOD SUPER->CONSULTANTIDSET\_GET\_ENTITYSET*  
*\*  EXPORTING*  
*\*    IV\_ENTITY\_NAME           =*  
*\*    IV\_ENTITY\_SET\_NAME       =*  
*\*    IV\_SOURCE\_NAME           =*  
*\*    IT\_FILTER\_SELECT\_OPTIONS =*  
*\*    IS\_PAGING                =*  
*\*    IT\_KEY\_TAB               =*  
*\*    IT\_NAVIGATION\_PATH       =*  
*\*    IT\_ORDER                 =*  
*\*    IV\_FILTER\_STRING         =*  
*\*    IV\_SEARCH\_STRING         =*  
*\*\*    io\_tech\_request\_context  =*  
*\*\*  IMPORTING*  
*\*\*    et\_entityset             =*  
*\*\*    es\_response\_context      =*  
*\*    .*  
*\*\*  CATCH /iwbep/cx\_mgw\_busi\_exception.*  
*\*\*  CATCH /iwbep/cx\_mgw\_tech\_exception.*  
*\*\*ENDTRY.*  
    DATA : r\_dispo  TYPE RANGE OF ztab\_gest\_cons-disponilbilty,  
           ls\_dispo LIKE LINE OF r\_dispo,  
           r\_expert  TYPE RANGE OF ztab\_gest\_cons-expertise,  
           ls\_expert LIKE LINE OF r\_expert.  
  
    DATA : ls\_filter\_select\_options TYPE /iwbep/s\_mgw\_select\_option,  
           lt\_select\_options        TYPE   /iwbep/t\_cod\_select\_options,  
           ls\_select\_options        TYPE  /iwbep/s\_cod\_select\_option.  
  
    READ TABLE it\_filter\_select\_options INTO ls\_filter\_select\_options WITH KEY property = 'Disponilbilty'.  
*\*    IF ls\_filter\_select\_options-property IS NOT INITIAL.*  
    IF sy-subrc eq 0.  
      lt\_select\_options[] =  ls\_filter\_select\_options-select\_options[].  
      LOOP AT lt\_select\_options INTO  ls\_select\_options.  
        ls\_dispo-sign = ls\_select\_options-sign.  
        ls\_dispo-option = ls\_select\_options-option.  
        ls\_dispo-low = ls\_select\_options-low.  
        ls\_dispo-high = ls\_select\_options-high.  
        APPEND ls\_dispo TO r\_dispo .  
        CLEAR : ls\_select\_options , ls\_dispo.  
      ENDLOOP.  
    ENDIF.  
  
    READ TABLE it\_filter\_select\_options INTO ls\_filter\_select\_options WITH KEY property = 'Expertise'.  
*\*    IF ls\_filter\_select\_options-property IS NOT INITIAL.*  
    IF sy-subrc eq 0.  
      lt\_select\_options[] =  ls\_filter\_select\_options-select\_options[].  
      LOOP AT lt\_select\_options INTO  ls\_select\_options.  
        ls\_expert-sign = ls\_select\_options-sign.  
        ls\_expert-option = ls\_select\_options-option.  
        ls\_expert-low = ls\_select\_options-low.  
        ls\_expert-high = ls\_select\_options-high.  
        APPEND ls\_expert TO r\_expert .  
        CLEAR : ls\_select\_options , ls\_expert.  
      ENDLOOP.  
    ENDIF.  
  
    SELECT \* FROM ztab\_gest\_cons  
      INTO CORRESPONDING FIELDS OF TABLE et\_entityset  
      WHERE disponilbilty IN r\_dispo  
      AND expertise IN r\_expert.  
  ENDMETHOD.