Report - ZAKASH\_TEST

*\*&---------------------------------------------------------------------\**  
*\*& Report  ZAKASH\_TEST*  
*\*&*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&*  
*\*&---------------------------------------------------------------------\**  
  
REPORT ZAKASH\_TEST.  
  
INCLUDE ztable\_backup\_top.  
  
INCLUDE ztable\_backup\_scr.  
  
INCLUDE ztable\_backup\_f01.  
  
INITIALIZATION.  
*\*&----------------------------------------------------------------------------------------\**  
*\*&        A T  S E L E C T I O N - S C R E E N    O U T P U T                             \**  
*\*&----------------------------------------------------------------------------------------\**  
AT SELECTION-SCREEN OUTPUT.  
  PERFORM at\_selection\_screen\_output.  
START-OF-SELECTION.  
  CASE gc\_x.  
    WHEN rb\_bu.  
      PERFORM f\_fetch\_data.  
    WHEN rb\_s\_bu.  
      PERFORM f\_specific\_data   USING p\_name.  
    WHEN rb\_r\_bu.  
      PERFORM f\_restore\_data CHANGING gv\_name.  
      PERFORM f\_read\_file.  
    WHEN rb\_d\_bu.  
      PERFORM f\_delete\_files.  
    WHEN rb\_ds\_bu.  
      PERFORM f\_delete\_spec\_files USING p\_name.  
    WHEN OTHERS.  
      *"Do Nothing*  
  ENDCASE.

Include - ZTABLE\_BACKUP\_TOP

*\*&---------------------------------------------------------------------\**  
*\*&  Include           ZTABLE\_BACKUP\_TOP*  
*\*&---------------------------------------------------------------------\**  
TYPE-POOLS : slis,icon.  
TYPES: BEGIN OF t\_file,  
       date        TYPE datum,  
       time        TYPE tims,  
       table\_name  TYPE tabname,  
       file\_name   TYPE string,  
       END OF t\_file,  
       BEGIN OF t\_user\_alv,  
       file\_count  TYPE i,  
       table\_count TYPE i,  
       END OF t\_user\_alv,  
       tt\_t\_file       TYPE STANDARD TABLE OF t\_file,  
       tt\_t\_user\_alv   TYPE STANDARD TABLE OF t\_user\_alv.  
DATA: gt\_app\_files     TYPE STANDARD TABLE OF epsfili,  
      gt\_fieldcatalog  TYPE TABLE OF slis\_fieldcat\_alv,  
      gt\_fieldcatalog2 TYPE STANDARD TABLE OF slis\_fieldcat\_alv,  
      gt\_file          TYPE STANDARD TABLE OF t\_file,  
      gwa\_file         TYPE t\_file,  
      gv\_name          TYPE fileintern,  
      gv\_count         TYPE i VALUE '0',  
      gv\_table\_count   TYPE i VALUE '0',  
      gv\_tabindex      TYPE sy-tabix,  
      gv\_file          TYPE string.  
CONSTANTS: gc\_x        TYPE char1 VALUE 'X'.

Include ZTABLE\_BACKUP\_SCR

*\*&---------------------------------------------------------------------\**  
*\*&  Include           ZTABLE\_BACKUP\_SCR*  
*\*&---------------------------------------------------------------------\**  
SELECTION-SCREEN BEGIN OF BLOCK a1 WITH FRAME TITLE text-001.  
PARAMETERS: rb\_bu     RADIOBUTTON GROUP g1 MODIF ID gr1 USER-COMMAND abc DEFAULT 'X',   *"Radio button 1*  
            rb\_s\_bu   RADIOBUTTON GROUP g1 MODIF ID gr1,                                *"Radio button 2*  
            rb\_r\_bu   RADIOBUTTON GROUP g1 MODIF ID gr1,                                *"Radio button 3*  
            rb\_d\_bu   RADIOBUTTON GROUP g1 MODIF ID gr1,                                *"Radio button 4*  
            rb\_ds\_bu  RADIOBUTTON GROUP g1 MODIF ID gr1.                                *"Radio button 5*  
SELECTION-SCREEN END OF BLOCK a1.  
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-002.  
PARAMETERS: p\_name TYPE tabname MATCHCODE OBJECT /bev3/chtabl MODIF ID gr2.*"Table Name*  
SELECTION-SCREEN END OF BLOCK b1.

Include ZTABLE\_BACKUP\_F01

*\*&---------------------------------------------------------------------\**  
*\*&  Include           ZTABLE\_BACKUP\_F01*  
*\*&---------------------------------------------------------------------\**  
FORM at\_selection\_screen\_output.  
*\*\* To provide the parameter for entering table name*  
  LOOP AT SCREEN.  
    IF rb\_bu             = 'X'.  
      IF screen-group1   = 'GR2'.  
        screen-invisible = 1.  
        screen-input     = 0 .  
        MODIFY SCREEN.  
      ENDIF.  
    ELSEIF rb\_r\_bu       = 'X'.  
      IF screen-group1   = 'GR2'.  
        screen-invisible = 1.  
        screen-input     = 0 .  
        MODIFY SCREEN.  
      ENDIF.  
    ELSEIF rb\_d\_bu       = 'X'.  
      IF screen-group1   = 'GR2'.  
        screen-invisible = 1.  
        screen-input     = 0 .  
        MODIFY SCREEN.  
      ENDIF.  
    ENDIF.  
  ENDLOOP.  
ENDFORM.                    *" AT\_SELECTION\_SCREEN\_OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_FETCH\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*       To fetch table contents*  
*\*----------------------------------------------------------------------\**  
FORM f\_fetch\_data.  
  DATA: lt\_tables        TYPE STANDARD TABLE OF ztable\_backup,  
        lwa\_tables       TYPE ztable\_backup,  
        ref\_table        TYPE REF TO data,  
        lv\_name          TYPE fileintern,  
        lv\_mess          TYPE string,  
        lv\_sys           TYPE sysysid.  
  CONSTANTS: lc\_x        TYPE char1      VALUE 'X',  
             lc\_log\_path TYPE fileintern VALUE 'Z\_TABLE\_BACKUP\_PATH'.  
  FIELD-SYMBOLS: <lt\_table\_content> TYPE STANDARD TABLE.  
*\* Fetching table names from configurable table*  
  SELECT   \*  
    FROM   ztable\_backup  
INTO TABLE lt\_tables  
   WHERE   zindicator EQ lc\_x.  
  LOOP AT lt\_tables INTO lwa\_tables.  
    CREATE DATA ref\_table TYPE TABLE OF (lwa\_tables-table\_name).  
    ASSIGN ref\_table->\* TO <lt\_table\_content>.  
    SELECT \*  
      FROM (lwa\_tables-table\_name)  
INTO TABLE <lt\_table\_content>  
     WHERE mandt EQ sy-mandt.  
    IF <lt\_table\_content> IS NOT INITIAL.  
      PERFORM f\_get\_file\_path USING lc\_log\_path  
                           CHANGING lv\_name.  
      IF lv\_name IS NOT INITIAL.  
        PERFORM f\_create\_file TABLES <lt\_table\_content>  
                               USING lv\_name  
                                     lwa\_tables-table\_name.  
      ENDIF.  
    ENDIF.  
    UNASSIGN <lt\_table\_content>.  
  ENDLOOP.  
  lv\_mess = 'Give Suitable message if required'.  
  lv\_sys  = sy-sysid.  
  CONDENSE lv\_sys.  
  REPLACE '&1' WITH lv\_sys INTO lv\_mess.  
  CONDENSE lv\_mess.  
  MESSAGE lv\_mess TYPE 'I'.  
ENDFORM.                    *" F\_FETCH\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_GET\_FILE\_PATH*  
*\*&---------------------------------------------------------------------\**  
*\*       Get Physical File Path*  
*\*----------------------------------------------------------------------\**  
FORM f\_get\_file\_path  USING xc\_log\_path TYPE fileintern  
                   CHANGING xv\_name     TYPE fileintern.  
  CLEAR: xv\_name.  
  CALL FUNCTION 'FILE\_GET\_NAME'  
    EXPORTING  
      logical\_filename = xc\_log\_path  
    IMPORTING  
      file\_name        = xv\_name  
    EXCEPTIONS  
      file\_not\_found   = 1  
      OTHERS           = 2.  
ENDFORM.                    *" F\_GET\_FILE\_PATH*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_CREATE\_FILE*  
*\*&---------------------------------------------------------------------\**  
*\*       To create file and upload on Application Server*  
*\*----------------------------------------------------------------------\**  
FORM f\_create\_file  TABLES xt\_table\_content TYPE STANDARD TABLE  
                    USING  xv\_name          TYPE fileintern  
                           xwa\_tablename    TYPE tabname.  
  DATA: lv\_outputfile     TYPE string,  
        lv\_outmsg         TYPE string,  
        lwa\_tables        TYPE ztable\_backup.  
  FIELD-SYMBOLS: <lfs\_table\_content> TYPE any.  
  CONSTANTS: lc\_slash     TYPE c     VALUE '/',  
             lc\_uscore    TYPE c     VALUE '\_',  
             lc\_dot       TYPE c     VALUE '.',  
             lc\_csv       TYPE char3 VALUE 'CSV',  
             lc\_sep       TYPE c     VALUE '-',  
             lc\_info      TYPE c     VALUE 'I',  
             lc\_pipe      TYPE c     VALUE '|'.  
  CONCATENATE xv\_name                 *"Path*  
              lc\_slash                *"/*  
              xwa\_tablename           *"Table Name*  
              lc\_pipe                 *"|*  
              sy-datum                *"YYYYMMDD*  
              lc\_pipe                 *"|*  
              sy-uzeit                *"HHMMSS*  
              lc\_dot                  *".*  
              lc\_csv                  *"CSV*  
         INTO lv\_outputfile.  
  OPEN DATASET lv\_outputfile FOR OUTPUT IN TEXT MODE  
                             ENCODING DEFAULT MESSAGE lv\_outmsg.  
  IF sy-subrc EQ 0.  
    LOOP AT xt\_table\_content ASSIGNING <lfs\_table\_content>.  
      TRANSFER <lfs\_table\_content> TO lv\_outputfile.  
    ENDLOOP.  
    CLOSE DATASET lv\_outputfile.  
  ELSE.  
    CONCATENATE lv\_outputfile lv\_outmsg INTO lv\_outmsg SEPARATED BY lc\_sep.  
    MESSAGE lv\_outmsg TYPE lc\_info. *"Output file open info*  
  ENDIF.  
ENDFORM.                    *" F\_CREATE\_FILE*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_SPECIFIC\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*       To fetch data of a specific table*  
*\*----------------------------------------------------------------------\**  
FORM f\_specific\_data USING x\_name TYPE tabname.  
  DATA: lwa\_tables       TYPE ztable\_backup,  
        lwa\_maintain     TYPE ztable\_backup,  
        lv\_name          TYPE fileintern,  
        lv\_mess          TYPE string,  
        lv\_sys           TYPE sysysid,  
        lv\_answer        TYPE char1,  
        ref\_table        TYPE REF TO data.  
  CONSTANTS: lc\_log\_path TYPE fileintern VALUE 'Z\_TABLE\_BACKUP\_PATH',  
             lc\_question TYPE char50     VALUE 'Table not in config. Select from below options',  
             lc\_op1      TYPE char20     VALUE 'Maintain & Backup',  
             lc\_op2      TYPE char20     VALUE 'Backup only'.  
  FIELD-SYMBOLS: <lt\_spec\_content> TYPE STANDARD TABLE.  
  SELECT SINGLE \*  
           FROM ztable\_backup  
           INTO lwa\_tables  
          WHERE table\_name EQ x\_name.  
  IF sy-subrc EQ 0.  
    CREATE DATA ref\_table TYPE TABLE OF (x\_name).  
    ASSIGN ref\_table->\* TO <lt\_spec\_content>.  
    SELECT \*  
      FROM (x\_name)  
INTO TABLE <lt\_spec\_content>  
     WHERE mandt EQ sy-mandt.  
    IF <lt\_spec\_content> IS NOT INITIAL.  
      PERFORM f\_get\_file\_path USING lc\_log\_path  
                           CHANGING lv\_name.  
      IF lv\_name IS NOT INITIAL.  
        PERFORM f\_create\_file TABLES <lt\_spec\_content>  
                               USING lv\_name  
                                     lwa\_tables-table\_name.  
      ENDIF.  
    ENDIF.  
    UNASSIGN <lt\_spec\_content>.  
    lv\_mess = text-019.  
    lv\_sys  = sy-sysid.  
    CONDENSE lv\_sys.  
    REPLACE '&1' WITH lv\_sys INTO lv\_mess.  
    CONDENSE lv\_mess.  
    MESSAGE lv\_mess TYPE 'I'.  
  ELSE.  
    CALL FUNCTION 'POPUP\_TO\_CONFIRM'  
      EXPORTING  
        text\_question = lc\_question  
        text\_button\_1 = lc\_op1  
        text\_button\_2 = lc\_op2  
      IMPORTING  
        answer        = lv\_answer.  
    IF lv\_answer EQ '1'.  
      lwa\_maintain-table\_name = x\_name.  
      lwa\_maintain-zindicator = 'X'.  
      INSERT ztable\_backup FROM lwa\_maintain.  
      IF sy-subrc EQ 0.  
        COMMIT WORK.  
      ENDIF.  
      CREATE DATA ref\_table TYPE TABLE OF (x\_name).  
      ASSIGN ref\_table->\* TO <lt\_spec\_content>.  
      SELECT \*  
        FROM (x\_name)  
  INTO TABLE <lt\_spec\_content>  
       WHERE mandt EQ sy-mandt.  
      IF <lt\_spec\_content> IS NOT INITIAL.  
        PERFORM f\_get\_file\_path USING lc\_log\_path  
                             CHANGING lv\_name.  
        IF lv\_name IS NOT INITIAL.  
          PERFORM f\_create\_file TABLES <lt\_spec\_content>  
                                 USING lv\_name  
                                       x\_name.  
        ENDIF.  
      ENDIF.  
      UNASSIGN <lt\_spec\_content>.  
      lv\_mess = text-019.  
      lv\_sys  = sy-sysid.  
      CONDENSE lv\_sys.  
      REPLACE '&1' WITH lv\_sys INTO lv\_mess.  
      CONDENSE lv\_mess.  
      MESSAGE lv\_mess TYPE 'I'.  
    ELSEIF lv\_answer EQ '2'.  
      CREATE DATA ref\_table TYPE TABLE OF (x\_name).  
      ASSIGN ref\_table->\* TO <lt\_spec\_content>.  
      SELECT \*  
        FROM (x\_name)  
  INTO TABLE <lt\_spec\_content>  
       WHERE mandt EQ sy-mandt.  
      IF <lt\_spec\_content> IS NOT INITIAL.  
        PERFORM f\_get\_file\_path USING lc\_log\_path  
                             CHANGING lv\_name.  
        IF lv\_name IS NOT INITIAL.  
          PERFORM f\_create\_file TABLES <lt\_spec\_content>  
                                 USING lv\_name  
                                       x\_name.  
        ENDIF.  
      ENDIF.  
      UNASSIGN <lt\_spec\_content>.  
      lv\_mess = text-019.  
      lv\_sys  = sy-sysid.  
      CONDENSE lv\_sys.  
      REPLACE '&1' WITH lv\_sys INTO lv\_mess.  
      CONDENSE lv\_mess.  
      MESSAGE lv\_mess TYPE 'I'.  
    ELSE.  
      EXIT.  
    ENDIF.  
  ENDIF.  
ENDFORM.                    *" F\_SPECIFIC\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_RESTORE\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*       To restore data*  
*\*----------------------------------------------------------------------\**  
FORM f\_restore\_data CHANGING xv\_name TYPE fileintern.  
  CONSTANTS: lc\_log\_path TYPE fileintern VALUE 'Z\_TABLE\_BACKUP\_PATH'.  
  PERFORM f\_get\_file\_path USING lc\_log\_path  
                       CHANGING xv\_name.  
  PERFORM f\_get\_files     USING xv\_name.  
ENDFORM.                    *" F\_RESTORE\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_GET\_FILES*  
*\*&---------------------------------------------------------------------\**  
*\*       To get all files from specified path*  
*\*----------------------------------------------------------------------\**  
FORM f\_get\_files  USING xv\_name TYPE fileintern.  
*\* Get all files at specified path.*  
  CALL FUNCTION 'EPS\_GET\_DIRECTORY\_LISTING'  
    EXPORTING  
      dir\_name             = xv\_name  
    TABLES  
      dir\_list             = gt\_app\_files  
    EXCEPTIONS  
      empty\_directory\_list = 1  
      OTHERS               = 2.  
ENDFORM.                    *" F\_GET\_FILES*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_READ\_FILE*  
*\*&---------------------------------------------------------------------\**  
*\*       To read files from application server*  
*\*----------------------------------------------------------------------\**  
FORM f\_read\_file.  
  DATA: lwa\_app\_files TYPE epsfili,  
        lwa\_file      TYPE t\_file,  
        lv\_file       TYPE string.  
  CONSTANTS: lc\_pipe TYPE c VALUE '|'.  
  LOOP AT gt\_app\_files INTO lwa\_app\_files.  
    lwa\_file-file\_name = lwa\_app\_files-name.  
    SPLIT lwa\_app\_files-name AT lc\_pipe INTO lwa\_file-table\_name lwa\_file-date lwa\_file-time.  
    APPEND lwa\_file TO gt\_file.  
    CLEAR: lwa\_file.  
  ENDLOOP.  
  IF gt\_file[] IS NOT INITIAL.  
    SORT gt\_file DESCENDING BY date time ASCENDING table\_name.  
  ENDIF.  
  PERFORM f\_create\_alv USING gt\_file gt\_fieldcatalog.  
ENDFORM.                    *" F\_READ\_FILE*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_CREATE\_ALV*  
*\*&---------------------------------------------------------------------\**  
*\*       To create ALV output display*  
*\*----------------------------------------------------------------------\**  
FORM f\_create\_alv USING xt\_file     TYPE tt\_t\_file  
                        xt\_fieldcat LIKE gt\_fieldcatalog.  
  DATA: lwa\_layout   TYPE  slis\_layout\_alv,  
        lt\_eventexit TYPE slis\_t\_event\_exit.  
  CLEAR: lwa\_layout.  
  lwa\_layout-colwidth\_optimize = 'X'.  
  PERFORM f\_fieldcatalog TABLES xt\_fieldcat.  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
    EXPORTING  
      i\_callback\_program      = sy-repid  
      i\_callback\_user\_command = 'USER\_COMMAND'  
      i\_callback\_top\_of\_page  = 'F\_TOP-OF-PAGE'  
      is\_layout               = lwa\_layout  
      it\_fieldcat             = xt\_fieldcat[]  
    TABLES  
      t\_outtab                = xt\_file[]  
    EXCEPTIONS  
      program\_error           = 1  
      OTHERS                  = 2.  
ENDFORM.                    *" F\_CREATE\_ALV*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_FIELDCATALOG*  
*\*&---------------------------------------------------------------------\**  
*\*       ALV Fieldcatalog*  
*\*----------------------------------------------------------------------\**  
FORM f\_fieldcatalog TABLES xt\_fieldcat LIKE gt\_fieldcatalog.  
  DATA: lwa\_fieldcatalog TYPE slis\_fieldcat\_alv,  
        lv\_col\_pos       TYPE sy-cucol.  
  CLEAR lv\_col\_pos.  
  DEFINE build\_fldcat.  
    clear lwa\_fieldcatalog.  
    lv\_col\_pos = lv\_col\_pos + 1.  
    lwa\_fieldcatalog-fieldname = &1.  
    lwa\_fieldcatalog-seltext\_m = &2.  
    lwa\_fieldcatalog-outputlen = &3.  
    lwa\_fieldcatalog-col\_pos   = lv\_col\_pos.  
    append lwa\_fieldcatalog to xt\_fieldcat.  
  END-OF-DEFINITION.  
  build\_fldcat text-005 text-006  8.    *"Date of Backup*  
  build\_fldcat text-020 text-021  6.    *"Time of Backup*  
  build\_fldcat text-003 text-004 20.    *"Table Name*  
  build\_fldcat text-012 text-013 30.    *"File Name*  
ENDFORM.                    *" F\_FIELDCATALOG*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  TOP-OF-PAGE*  
*\*&---------------------------------------------------------------------\**  
*\*&       Top of page*  
*\*&---------------------------------------------------------------------\**  
FORM f\_top-of-page.  
*\*\* ALV Header declarations*  
  DATA: lt\_header     TYPE slis\_t\_listheader,  
        lwa\_header    TYPE slis\_listheader,  
        lv\_count      TYPE i.  
  lwa\_header-typ = text-007.  
  lwa\_header-key = text-008.  
  lwa\_header-typ = text-009.  
  lwa\_header-info = text-010.  
  APPEND lwa\_header TO lt\_header.  
  CLEAR lwa\_header.  
  CONCATENATE sy-datum+6(2) '.'  
              sy-datum+4(2) '.'  
              sy-datum(4) INTO lwa\_header-info. *"Input date*  
  APPEND lwa\_header TO lt\_header.  
  CLEAR: lwa\_header.  
  DESCRIBE TABLE gt\_app\_files LINES lv\_count.  
  lwa\_header-typ = text-007.  
  lwa\_header-key = text-011.  
  lwa\_header-info = lv\_count.  
  APPEND lwa\_header TO lt\_header.  
  CLEAR: lwa\_header.  
  CALL FUNCTION 'REUSE\_ALV\_COMMENTARY\_WRITE'  
    EXPORTING  
      it\_list\_commentary = lt\_header.  
ENDFORM.                    *"f\_top-of-page*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  user\_command*  
*\*&---------------------------------------------------------------------\**  
*\*       User Actions*  
*\*----------------------------------------------------------------------\**  
*\*User actions on ALV*  
FORM user\_command USING r\_ucomm     TYPE sy-ucomm  
                        rs\_selfield TYPE slis\_selfield.  
  DATA: lt\_file  TYPE STANDARD TABLE OF t\_user\_alv,  
        lwa\_file TYPE t\_user\_alv.  
  CLEAR: gv\_count, gv\_table\_count.  
  REFRESH: gt\_fieldcatalog2.  
  CASE r\_ucomm.  
    WHEN '&IC1'.  
      PERFORM f\_user\_command USING rs\_selfield.  
      lwa\_file-file\_count  = gv\_count.  
      lwa\_file-table\_count = gv\_table\_count.  
      APPEND lwa\_file TO lt\_file.  
      CLEAR: lwa\_file.  
      PERFORM f\_alv USING lt\_file  
                          gt\_fieldcatalog2.  
  ENDCASE.  
ENDFORM.                    *"user\_command*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_USER\_COMMAND*  
*\*&---------------------------------------------------------------------\**  
*\*       To process user command*  
*\*----------------------------------------------------------------------\**  
FORM f\_user\_command USING rs\_selfield TYPE slis\_selfield.  
  DATA: lv\_date        TYPE datum,  
        lwa\_data       TYPE string,  
        ref\_table      TYPE REF TO data.  
  CONSTANTS: lc\_slash  TYPE c     VALUE '/',  
             lc\_uscore TYPE c     VALUE '\_'.  
  FIELD-SYMBOLS: <lt\_count> TYPE STANDARD TABLE.  
  READ TABLE gt\_file INTO gwa\_file INDEX rs\_selfield-tabindex.  
  IF sy-subrc EQ 0.  
    CONCATENATE gv\_name lc\_slash gwa\_file-file\_name INTO gv\_file.  
    OPEN DATASET gv\_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
    IF sy-subrc EQ 0.  
      DO.  
        READ DATASET gv\_file INTO lwa\_data.  
        IF sy-subrc EQ 0.  
          gv\_count = gv\_count + 1.  
        ELSE.  
          EXIT.  
        ENDIF.  
      ENDDO.  
    ENDIF.  
    CLOSE DATASET gv\_file.  
  ENDIF.  
  CREATE DATA ref\_table TYPE TABLE OF (gwa\_file-table\_name).  
  ASSIGN ref\_table->\* TO <lt\_count>.  
  SELECT \*  
    FROM (gwa\_file-table\_name)  
    INTO TABLE <lt\_count>  
   WHERE mandt EQ sy-mandt.  
  IF sy-subrc EQ 0.  
    DESCRIBE TABLE <lt\_count> LINES gv\_table\_count.  
  ELSE.  
    gv\_table\_count = 0.  
  ENDIF.  
ENDFORM.                    *" F\_USER\_COMMAND*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_ALV*  
*\*&---------------------------------------------------------------------\**  
*\*       User command ALV*  
*\*----------------------------------------------------------------------\**  
FORM f\_alv USING xt\_file     TYPE tt\_t\_user\_alv  
                 xt\_fieldcat LIKE gt\_fieldcatalog2.  
  DATA: lwa\_layout   TYPE  slis\_layout\_alv,  
        lt\_eventexit TYPE slis\_t\_event\_exit.  
  CLEAR: lwa\_layout.  
  lwa\_layout-colwidth\_optimize = 'X'.  
  PERFORM f\_fieldcatalog2 TABLES xt\_fieldcat.  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
    EXPORTING  
      i\_callback\_program       = sy-repid  
      i\_callback\_pf\_status\_set = 'PFSTATUS'  
      i\_callback\_user\_command  = 'USER\_COMMAND\_2'  
      is\_layout                = lwa\_layout  
      it\_fieldcat              = xt\_fieldcat[]  
    TABLES  
      t\_outtab                 = xt\_file[]  
    EXCEPTIONS  
      program\_error            = 1  
      OTHERS                   = 2.  
ENDFORM.                    *" F\_ALV*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_FIELDCATALOG2*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
*\*      -->P\_XT\_FIELDCAT  text*  
*\*----------------------------------------------------------------------\**  
FORM f\_fieldcatalog2 TABLES xt\_fieldcat LIKE gt\_fieldcatalog2.  
  DATA: lwa\_fieldcatalog TYPE slis\_fieldcat\_alv,  
        lv\_col\_pos       TYPE sy-cucol.  
  CLEAR lv\_col\_pos.  
  DEFINE build\_fldcat.  
    clear lwa\_fieldcatalog.  
    lv\_col\_pos = lv\_col\_pos + 1.  
    lwa\_fieldcatalog-fieldname = &1.  
    lwa\_fieldcatalog-seltext\_m = &2.  
    lwa\_fieldcatalog-col\_pos   = lv\_col\_pos.  
    append lwa\_fieldcatalog to xt\_fieldcat.  
  END-OF-DEFINITION.  
  build\_fldcat text-016 text-014.       *"File Data Count*  
  build\_fldcat text-017 text-015.       *"Table Data Count*  
ENDFORM.                    *" F\_FIELDCATALOG2*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  PF\_STATUS\_SET*  
*\*&---------------------------------------------------------------------\**  
FORM pfstatus USING lt\_extab TYPE kkblo\_t\_extab.  
*\*--- Excluding table*  
  TYPES: kkblo\_t\_extab TYPE kkblo\_extab OCCURS 1.  
  SET PF-STATUS 'STANDARD' EXCLUDING lt\_extab.  
ENDFORM.                    *" PF\_STATUS\_SET*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  user\_command*  
*\*&---------------------------------------------------------------------\**  
*\*       User Actions*  
*\*----------------------------------------------------------------------\**  
*\*User actions on ALV*  
FORM user\_command\_2 USING r\_ucomm     TYPE sy-ucomm  
                          rs\_selfield TYPE slis\_selfield.  
  DATA: lt\_file  TYPE STANDARD TABLE OF t\_user\_alv,  
        lwa\_file TYPE t\_user\_alv.  
  CASE r\_ucomm.  
    WHEN 'OVERWRITE'.  
      PERFORM f\_overwrite\_table USING r\_ucomm.  
    WHEN 'INSERT'.  
      PERFORM f\_overwrite\_table USING r\_ucomm.  
    WHEN 'CANCEL'.  
      MESSAGE text-018 TYPE 'I'.  
      LEAVE TO SCREEN 0.  
  ENDCASE.  
ENDFORM.                    *"user\_command*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_OVERWRITE\_TABLE*  
*\*&---------------------------------------------------------------------\**  
*\*       To overwrite table with file data*  
*\*----------------------------------------------------------------------\**  
FORM f\_overwrite\_table USING r\_ucomm TYPE sy-ucomm.  
  FIELD-SYMBOLS: <lfs\_data>     TYPE any,  
                 <fs>           TYPE any,  
                 <lfs\_data\_tab> TYPE STANDARD TABLE.  
  DATA: ref\_table TYPE REF TO data,  
        ref\_tab   TYPE REF TO data,  
        lv\_bef\_count TYPE i,  
        lv\_aft\_count TYPE i,  
        lv\_count     TYPE i,  
        lv\_mess      TYPE string,  
        lv\_cou       TYPE string.  
  CREATE DATA ref\_table TYPE TABLE OF (gwa\_file-table\_name).  
  ASSIGN ref\_table->\* TO <lfs\_data\_tab>.  
  CREATE DATA ref\_tab TYPE (gwa\_file-table\_name).  
  ASSIGN ref\_tab->\* TO <lfs\_data>.  
  IF r\_ucomm EQ 'OVERWRITE'.  
    DELETE FROM (gwa\_file-table\_name).  
    IF sy-subrc EQ 0.  
      COMMIT WORK.  
    ENDIF.  
  ELSEIF r\_ucomm EQ 'INSERT'.  
    SELECT COUNT(\*)  
            FROM (gwa\_file-table\_name)  
            INTO lv\_bef\_count  
           WHERE mandt EQ sy-mandt.  
  ENDIF.  
  OPEN DATASET gv\_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
  IF sy-subrc EQ 0.  
    DO.  
      READ DATASET gv\_file INTO <lfs\_data>.  
      IF sy-subrc EQ 0.  
        APPEND <lfs\_data> TO <lfs\_data\_tab>.  
      ELSE.  
        EXIT.  
      ENDIF.  
    ENDDO.  
    MODIFY (gwa\_file-table\_name) FROM TABLE <lfs\_data\_tab>.  
    IF sy-subrc EQ 0.  
      COMMIT WORK.  
      IF r\_ucomm EQ 'OVERWRITE'.  
        MESSAGE text-024 TYPE 'I'.  
      ELSEIF r\_ucomm EQ 'INSERT'.  
        SELECT COUNT(\*)  
                FROM (gwa\_file-table\_name)  
                INTO lv\_aft\_count  
               WHERE mandt EQ sy-mandt.  
        lv\_count = lv\_aft\_count - lv\_bef\_count.  
        lv\_cou   = lv\_count.  
        lv\_mess = text-025.  
        REPLACE '&1' WITH lv\_cou INTO lv\_mess.  
        CONDENSE lv\_mess.  
        MESSAGE lv\_mess TYPE 'I'.  
      ENDIF.  
    ENDIF.  
  ENDIF.  
  CLOSE DATASET gv\_file.  
ENDFORM.                    *" F\_OVERWRITE\_TABLE*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_DELETE\_FILES*  
*\*&---------------------------------------------------------------------\**  
*\*       To delete files from application server*  
*\*----------------------------------------------------------------------\**  
FORM f\_delete\_files.  
  DATA: lv\_date\_conf     TYPE tvarv\_val,  
        lv\_name          TYPE fileintern,  
        lv\_date          TYPE datum,  
        lv\_table\_name    TYPE tabname,  
        lv\_dat           TYPE datum,  
        lv\_file          TYPE string,  
        lv\_mess          TYPE string,  
        lwa\_app\_files    TYPE epsfili.  
  CONSTANTS: lc\_log\_path TYPE fileintern VALUE 'Z\_TABLE\_BACKUP\_PATH',  
             lc\_uscore   TYPE c          VALUE '\_',  
             lc\_slash    TYPE c          VALUE '/'.  
  SELECT SINGLE low  
    FROM tvarvc  
    INTO lv\_date\_conf  
   WHERE name EQ 'ZDAT\_BACKUP'  
     AND type EQ 'P'.  
  IF sy-subrc EQ 0.  
    lv\_dat = sy-datum - lv\_date\_conf.  
  ENDIF.  
  PERFORM f\_get\_file\_path USING lc\_log\_path  
                       CHANGING lv\_name.  
  PERFORM f\_get\_files USING lv\_name.  
  LOOP AT gt\_app\_files INTO lwa\_app\_files.  
    SPLIT lwa\_app\_files-name AT lc\_uscore INTO lv\_table\_name lv\_date.  
    IF lv\_date LT lv\_dat.  
      CONCATENATE lv\_name lc\_slash lwa\_app\_files-name INTO lv\_file.  
      OPEN DATASET lv\_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
      DELETE DATASET lv\_file.  
    ENDIF.  
  ENDLOOP.  
  lv\_mess = text-022.  
  REPLACE '&1' WITH lv\_date\_conf INTO lv\_mess.  
  CONDENSE lv\_mess.  
  MESSAGE lv\_mess TYPE 'I'.  
ENDFORM.                    *" F\_DELETE\_FILES*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_DELETE\_SPEC\_FILES*  
*\*&---------------------------------------------------------------------\**  
*\*       To delete spec table backup files*  
*\*----------------------------------------------------------------------\**  
FORM f\_delete\_spec\_files USING x\_name TYPE tabname.  
  DATA:      lv\_name          TYPE fileintern,  
             lwa\_app\_files    TYPE epsfili,  
             lv\_date          TYPE datum,  
             lv\_table\_name    TYPE tabname,  
             lv\_file          TYPE string,  
             lv\_mess          TYPE string,  
             lv\_flag          TYPE c.  
  CONSTANTS: lc\_log\_path TYPE fileintern VALUE 'Z\_TABLE\_BACKUP\_PATH',  
             lc\_slash    TYPE c          VALUE '/',  
             lc\_pipe     TYPE c          VALUE '|'.  
  PERFORM f\_get\_file\_path USING lc\_log\_path  
                       CHANGING lv\_name.  
  PERFORM f\_get\_files USING lv\_name.  
  LOOP AT gt\_app\_files INTO lwa\_app\_files.  
    SPLIT lwa\_app\_files-name AT lc\_pipe INTO lv\_table\_name lv\_date.  
    IF lv\_table\_name EQ x\_name.  
      CONCATENATE lv\_name lc\_slash lwa\_app\_files-name INTO lv\_file.  
      OPEN DATASET lv\_file FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
      DELETE DATASET lv\_file.  
      lv\_flag = abap\_true.  
    ENDIF.  
  ENDLOOP.  
  IF lv\_flag EQ abap\_true.  
    lv\_mess = text-023.  
    REPLACE '&1' WITH x\_name INTO lv\_mess.  
    CONDENSE lv\_mess.  
    MESSAGE lv\_mess TYPE 'I'.  
  ELSE.  
    MESSAGE text-026 TYPE 'I'.  
  ENDIF.  
ENDFORM.                    *" F\_DELETE\_SPEC\_FILES*