YBC\_USER\_STATISTICS\_ALV

USER 1 Types SYUNAME

TABLENAME 1 Types YTABLENAME

OBJECT\_DESC 1 Types YOBJECT\_DESC

OBJECT 1 Types YOBJECT

OBJECT2 1 Types YOBJECT

OBJECT3 1 Types YOBJECT

DATE 1 Types DATS

TIME 1 Types UZEIT

COUNTER 1 Types

NAME\_FIRST 1 Types AD\_NAMEFIR

NAME\_LAST 1 Types AD\_NAMELAS

\*&---------------------------------------------------------------------\*  
\*& Report YBC\_USER\_STATISTICS  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT ybc\_user\_statistics.  
\*{   INSERT         DLSK900088                                        1  
  
TYPE-POOLS: slis.  
  
INCLUDE ybc\_user\_statistics\_alv\_data.

\*&---------------------------------------------------------------------\*  
\*&  Include           YBC\_USER\_STATISTICS\_ALV\_DATA  
\*&---------------------------------------------------------------------\*  
  
  
TYPE-POOLS: slis,  
            icon.  
  
\* Data to be displayed --> create Structure in DDIC (SE11)  
DATA: gt\_data TYPE TABLE OF YBC\_USER\_STATISTICS\_ALV WITH HEADER LINE,  
      gs\_data TYPE YBC\_USER\_STATISTICS\_ALV.  
DATA: ok\_code  LIKE syst-ucomm,  
      g\_repid  LIKE sy-repid,  
      g\_save,  
      gv\_index LIKE sy-tabix.  
  
\* Containers & Grids  
DATA: my\_container  TYPE REF TO cl\_gui\_docking\_container.  
  
DATA: my\_grid       TYPE REF TO cl\_gui\_alv\_grid.  
  
DATA: it\_events TYPE cntl\_simple\_events.  
DATA: wa\_events LIKE LINE OF it\_events.  
  
DATA: it\_header TYPE    TABLE OF slis\_listheader WITH HEADER LINE,  
      it\_layout\_tree    TYPE lvc\_s\_layo,  
      it\_fieldcat\_lvc   TYPE lvc\_t\_fcat WITH HEADER LINE,  
      it\_fieldcat\_order TYPE lvc\_t\_fcat WITH HEADER LINE,  
      wa\_fieldcatalog   TYPE lvc\_s\_fcat,  
      it\_sort           TYPE lvc\_t\_sort, "Sortiertabelle  
      gs\_layout\_alv     TYPE lvc\_s\_layo,  
      gt\_sort\_grid      TYPE lvc\_t\_sort WITH HEADER LINE,  
      it\_icons          TYPE lvc\_t\_fimg,  
      wa\_icons LIKE LINE OF it\_icons,  
      it\_variant        TYPE disvariant,  
      it\_ord\_variant    TYPE disvariant,  
      x\_layout          TYPE disvariant,  
      x\_save,         "for parameter I\_SAVE: modus for saving a layout  
      g\_exit      TYPE c, "set if the user has aborted a layout popup  
      spec\_layout TYPE disvariant.    "specific layout  
INCLUDE ybc\_user\_statistics\_alv\_events.  
\*&---------------------------------------------------------------------\*  
\*&  Include           YBC\_USER\_STATISTICS\_ALV\_EVENTS  
\*&---------------------------------------------------------------------\*  
  
CLASS lcl\_event\_handler DEFINITION.  
  
  PUBLIC SECTION.  
\*   hotspot click trigger  
    METHODS hotspot\_click  
      FOR EVENT hotspot\_click OF cl\_gui\_alv\_grid  
      IMPORTING e\_row\_id  
                e\_column\_id  
                es\_row\_no.  
  
\*   Handle toolbar  
    METHODS handle\_toolbar  
      FOR EVENT toolbar OF cl\_gui\_alv\_grid  
      IMPORTING e\_object e\_interactive.  
  
\*   Handle user command  
    METHODS handle\_user\_command  
      FOR EVENT user\_command OF cl\_gui\_alv\_grid  
      IMPORTING e\_ucomm.  
  
  PRIVATE SECTION.  
    DATA: dialogbox\_status TYPE c.  "'X': does exist, SPACE: does not ex  
  
ENDCLASS.                    "lcl\_event\_handler DEFINITION  
  
\*\*---------------------------------------------------------------------\*  
\*\*       CLASS CL\_TREE\_EVENT\_RECEIVER IMPLEMENTATION  
\*\*---------------------------------------------------------------------\*  
CLASS lcl\_event\_handler  IMPLEMENTATION.  
\* handle trigger hotspot click  
  METHOD hotspot\_click.  
    DATA: rs\_selfield TYPE slis\_selfield.  
    rs\_selfield-tabindex = es\_row\_no-row\_id.  
    rs\_selfield-fieldname = e\_column\_id.  
    PERFORM user\_hot\_spot USING rs\_selfield.  
  ENDMETHOD.                    "hotspot\_click  
  
\* Handle toolbar  
  METHOD handle\_toolbar.  
    DATA: ls\_toolbar TYPE stb\_button.  
\* Append separator  
    CLEAR ls\_toolbar.  
    MOVE 3 TO ls\_toolbar-butn\_type.  
    APPEND ls\_toolbar TO e\_object->mt\_toolbar.  
  
\* Append Refresh  
    CLEAR ls\_toolbar.  
    MOVE 'REFRESH'                         TO ls\_toolbar-function.  
    MOVE icon\_refresh   TO ls\_toolbar-icon.  
    MOVE 'Refresh'(113) TO ls\_toolbar-quickinfo.  
    MOVE 'Refresh'(114) TO ls\_toolbar-text.  
    MOVE ' '                           TO ls\_toolbar-disabled.  
    APPEND ls\_toolbar                  TO e\_object->mt\_toolbar.  
  
  ENDMETHOD.                    "handle\_toolbar  
  
\* Handle user\_command  
  METHOD handle\_user\_command.  
    DATA: t\_rows TYPE lvc\_t\_row.  
    DATA: s\_row  TYPE lvc\_s\_row.  
  
    CASE e\_ucomm.  
      WHEN 'REFRESH'.  
        PERFORM get\_data.  
        CALL METHOD my\_grid->refresh\_table\_display  
          EXCEPTIONS  
            finished = 1  
            OTHERS   = 2.  
    ENDCASE.  
  ENDMETHOD.                    "handle\_user\_command  
  
  
ENDCLASS.                    "lcl\_event\_handler IMPLEMENTATION  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  user\_hot\_spot  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
FORM user\_hot\_spot USING p\_rs\_selfield TYPE slis\_selfield.  
  CLEAR: gs\_data.  
  READ TABLE gt\_data INDEX p\_rs\_selfield-tabindex INTO gs\_data.  
  CHECK NOT gs\_data IS INITIAL.  
  CASE p\_rs\_selfield-fieldname.  
    WHEN 'OBJECT'.  
      IF gs\_data-object IS NOT INITIAL.  
        CASE gs\_data-tablename.  
          WHEN 'EKKO'.  
            SET PARAMETER ID 'BES' FIELD gs\_data-object.  
            CALL TRANSACTION 'ME23N' AND SKIP FIRST SCREEN.  
          WHEN 'VBAK'.  
            SET PARAMETER ID 'AUN' FIELD gs\_data-object.  
            CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.  
          WHEN 'VBRK'.  
            SET PARAMETER ID 'VF' FIELD gs\_data-object.  
            CALL TRANSACTION 'VF03' AND SKIP FIRST SCREEN.  
          WHEN 'LIKP'.  
            SET PARAMETER ID 'VL' FIELD gs\_data-object.  
            CALL TRANSACTION 'VL03N' AND SKIP FIRST SCREEN.  
          WHEN 'VTTK'.  
            SET PARAMETER ID 'TNR' FIELD gs\_data-object.  
            CALL TRANSACTION 'VT03N' AND SKIP FIRST SCREEN.  
          WHEN 'CAUFV'.  
            SET PARAMETER ID 'ANR' FIELD gs\_data-object.  
            CALL TRANSACTION 'CO03' AND SKIP FIRST SCREEN.  
          WHEN 'CAUFV2'.  
            SET PARAMETER ID 'ANR' FIELD gs\_data-object.  
            CALL TRANSACTION 'IW33' AND SKIP FIRST SCREEN.  
          WHEN 'CAUFV3'.  
            SET PARAMETER ID 'ANR' FIELD gs\_data-object.  
            CALL TRANSACTION 'CN23' AND SKIP FIRST SCREEN.  
          WHEN 'PRPS'.  
            SET PARAMETER ID 'PRO' FIELD gs\_data-object.  
            CALL TRANSACTION 'CJ03' AND SKIP FIRST SCREEN.  
          WHEN 'BKPF'.  
            SET PARAMETER ID 'BLN' FIELD gs\_data-object.  
            CALL TRANSACTION 'FB03' AND SKIP FIRST SCREEN.  
\*        when 'EKET'.  
\*          SET PARAMETER ID 'VRT' FIELD gs\_data-object.  
\*          CALL TRANSACTION 'ME33L' AND SKIP FIRST SCREEN.  
          WHEN 'EBAN'.  
            SET PARAMETER ID 'BAN' FIELD gs\_data-object.  
            CALL TRANSACTION 'ME53N' AND SKIP FIRST SCREEN.  
          WHEN 'MARA'.  
            SET PARAMETER ID 'MAT' FIELD gs\_data-object.  
            CALL TRANSACTION 'MM03' AND SKIP FIRST SCREEN.  
          WHEN 'CRHD'.  
            SET PARAMETER ID 'AGR' FIELD gs\_data-object.  
            SET PARAMETER ID 'WRK' FIELD gs\_data-object2.  
            CALL TRANSACTION 'CR03' AND SKIP FIRST SCREEN.  
          WHEN 'STKO'.  
            SET PARAMETER ID 'MAT' FIELD gs\_data-object.  
            SET PARAMETER ID 'CSV' FIELD '1'.  
            CALL TRANSACTION 'CS03' AND SKIP FIRST SCREEN.  
          WHEN 'MKPF'.  
            SET PARAMETER ID 'MBN' FIELD gs\_data-object.  
            SET PARAMETER ID 'MJA' FIELD gs\_data-object2.  
            CALL TRANSACTION 'MB03' AND SKIP FIRST SCREEN.  
          WHEN 'LAGP'.  
            SET PARAMETER ID 'LGP' FIELD gs\_data-object.  
            SET PARAMETER ID 'LGT' FIELD gs\_data-object2.  
            SET PARAMETER ID 'LGN' FIELD gs\_data-object3.  
            CALL TRANSACTION 'LS03' AND SKIP FIRST SCREEN.  
          WHEN 'LTAK'.  
            SET PARAMETER ID 'TAN' FIELD gs\_data-object.  
            SET PARAMETER ID 'LGN' FIELD gs\_data-object2.  
            CALL TRANSACTION 'LT21' AND SKIP FIRST SCREEN.  
  
          WHEN 'KNA1'.  
            SET PARAMETER ID 'KUN' FIELD gs\_data-object.  
            CALL TRANSACTION 'XD03' AND SKIP FIRST SCREEN.  
  
          WHEN 'LFA1'.  
            DATA: kdy\_val(8) VALUE '/110'.  
            SET PARAMETER ID 'LIF' FIELD gs\_data-object.  
            SET PARAMETER ID 'KDY' FIELD kdy\_val.  
            CALL TRANSACTION 'XK03' AND SKIP FIRST SCREEN.  
  
          WHEN 'KONH'.  
            "do nothing  
  
        ENDCASE.  
      ENDIF.  
    WHEN 'USER'.  
      SET PARAMETER ID 'XUS' FIELD gs\_data-user.  
      CALL TRANSACTION 'SU01' AND SKIP FIRST SCREEN.  
  ENDCASE.  
ENDFORM.                               " USER\_COMMAND  
  
DATA: event\_handler TYPE REF TO lcl\_event\_handler.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* SELECTION SCREEN                                               \*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Selections for data  
SELECTION-SCREEN BEGIN OF BLOCK 1 WITH FRAME TITLE TEXT-001.  
  SELECT-OPTIONS:  s\_user   FOR sy-uname MATCHCODE OBJECT user\_comp OBLIGATORY,  
                   s\_datum  FOR sy-datum.  
  PARAMETERS: p\_vbak   AS CHECKBOX DEFAULT 'X',  
              p\_likp   AS CHECKBOX DEFAULT 'X',  
              p\_vbrk   AS CHECKBOX DEFAULT 'X',  
              p\_vttk   AS CHECKBOX DEFAULT 'X',  
              p\_mara   AS CHECKBOX DEFAULT 'X',  
              p\_cabn   AS CHECKBOX DEFAULT 'X',  
              p\_crhd   AS CHECKBOX DEFAULT 'X',  
              p\_plko   AS CHECKBOX DEFAULT 'X',  
              p\_stko   AS CHECKBOX DEFAULT 'X',  
              p\_afru   AS CHECKBOX DEFAULT 'X',  
              p\_mkpf   AS CHECKBOX DEFAULT 'X',  
              p\_caufv  AS CHECKBOX DEFAULT 'X',  
              p\_caufv2 AS CHECKBOX DEFAULT 'X',  
              p\_caufv3 AS CHECKBOX DEFAULT 'X',  
              p\_eban   AS CHECKBOX DEFAULT 'X',  
              p\_ekko   AS CHECKBOX DEFAULT 'X',  
\*            p\_eket   as checkbox default 'X',  
              p\_lagp   AS CHECKBOX DEFAULT 'X',  
              p\_ltak   AS CHECKBOX DEFAULT 'X',  
              p\_vekp   AS CHECKBOX DEFAULT 'X',  
              p\_kna1   AS CHECKBOX DEFAULT 'X',  
              p\_lfa1   AS CHECKBOX DEFAULT 'X',  
              p\_konh   AS CHECKBOX DEFAULT 'X',  
              p\_pspnr  AS CHECKBOX DEFAULT 'X',  
              p\_belnr  AS CHECKBOX DEFAULT 'X'.  
  
  
  
  
  
  
SELECTION-SCREEN END OF BLOCK 1.  
  
  
\* Selections for ALV  
SELECTION-SCREEN BEGIN OF BLOCK 3 WITH FRAME TITLE TEXT-002.  
\* Display Variant  
  PARAMETERS: p\_vari LIKE disvariant-variant.  
SELECTION-SCREEN END OF BLOCK 3.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* INITIALIZATION                                                 \*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
INITIALIZATION.  
  g\_repid          = sy-repid.  
  
  CLEAR spec\_layout.  
  MOVE g\_repid TO spec\_layout-report.  
  CALL FUNCTION 'LVC\_VARIANT\_DEFAULT\_GET'  
    EXPORTING  
      i\_save     = 'A'  
    CHANGING  
      cs\_variant = spec\_layout  
    EXCEPTIONS  
      not\_found  = 2.  
  IF sy-subrc = 0.  
    p\_vari = spec\_layout-variant.  
  ENDIF.  
  
\* Set Options: save variants userspecific or general  
  g\_save = 'A'.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* AT SELECTION-SCREEN  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_vari.  
\* popup F4 help to select a layout  
  CLEAR x\_layout.  
  MOVE g\_repid TO x\_layout-report.  
  CALL FUNCTION 'LVC\_VARIANT\_F4'  
    EXPORTING  
      is\_variant = x\_layout  
      i\_save     = x\_save  
    IMPORTING  
      e\_exit     = g\_exit  
      es\_variant = spec\_layout  
    EXCEPTIONS  
      not\_found  = 1  
      OTHERS     = 2.  
  IF sy-subrc NE 0.  
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
            WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ELSE.  
    IF g\_exit NE 'X'.  
\* set name of layout on selection screen  
      p\_vari    = spec\_layout-variant.  
    ENDIF.  
  ENDIF.  
  
AT SELECTION-SCREEN.  
\* test if specified layout exist  
  IF NOT p\_vari IS INITIAL.  
    CLEAR spec\_layout.  
    MOVE p\_vari  TO spec\_layout-variant.  
    MOVE g\_repid TO spec\_layout-report.  
  
    CALL FUNCTION 'LVC\_VARIANT\_EXISTENCE\_CHECK'  
      EXPORTING  
        i\_save        = x\_save  
      CHANGING  
        cs\_variant    = spec\_layout  
      EXCEPTIONS  
        wrong\_input   = 1  
        not\_found     = 2  
        program\_error = 3  
        OTHERS        = 4.  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno  
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\* START OF SELECTION                                               \*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
START-OF-SELECTION.  
  
  PERFORM get\_data.  
  
  g\_repid = sy-repid.  
  CALL SCREEN '9000'.  
  
END-OF-SELECTION.  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  get\_data  
\*&---------------------------------------------------------------------\*  
FORM get\_data.  
  
  CLEAR: gt\_data, gs\_data.  
  REFRESH: gt\_data.  
  
\*VBAK  
  IF p\_vbak EQ 'X'.  
    DATA: lt\_vbak TYPE TABLE OF vbak.  
    DATA: ls\_vbak TYPE vbak.  
    REFRESH lt\_vbak.  
    SELECT vbeln ernam erdat erzet FROM vbak  
      INTO CORRESPONDING FIELDS OF TABLE lt\_vbak  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_vbak INTO ls\_vbak.  
      CLEAR gs\_data.  
      MOVE ls\_vbak-vbeln TO gs\_data-object.  
      MOVE ls\_vbak-ernam TO gs\_data-user.  
      MOVE ls\_vbak-erdat TO gs\_data-date.  
      MOVE ls\_vbak-erzet TO gs\_data-time.  
      MOVE 'VBAK'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*BKPF  
  IF p\_belnr EQ 'X'.  
    DATA: lt\_bkpf TYPE TABLE OF bkpf.  
    DATA: ls\_bkpf TYPE bkpf.  
    REFRESH lt\_bkpf.  
    SELECT belnr usnam cpudt  FROM bkpf  
      INTO CORRESPONDING FIELDS OF TABLE lt\_bkpf  
      WHERE usnam IN s\_user  
        AND cpudt IN s\_datum.  
  
    LOOP AT lt\_bkpf INTO ls\_bkpf.  
      CLEAR gs\_data.  
      MOVE ls\_bkpf-belnr TO gs\_data-object.  
      MOVE ls\_bkpf-usnam TO gs\_data-user.  
      MOVE ls\_bkpf-cpudt TO gs\_data-date.  
      MOVE 'BKPF'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*PRPS  
  IF p\_pspnr EQ 'X'.  
    DATA: lt\_prps TYPE TABLE OF prps.  
    DATA: ls\_prps TYPE prps.  
    REFRESH lt\_prps.  
    SELECT posid ernam erdat FROM prps  
      INTO CORRESPONDING FIELDS OF TABLE lt\_prps  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_prps INTO ls\_prps.  
      CLEAR gs\_data.  
      MOVE ls\_prps-posid TO gs\_data-object.  
      MOVE ls\_prps-ernam TO gs\_data-user.  
      MOVE ls\_prps-erdat TO gs\_data-date.  
      MOVE 'PRPS'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*AUFK  
\*  IF p\_aufnr EQ 'X'.  
\*    DATA: lt\_aufk TYPE TABLE OF aufk.  
\*    DATA: ls\_aufk TYPE aufk.  
\*    REFRESH lt\_aufk.  
\*    SELECT aufnr ernam erdat FROM aufk  
\*      INTO CORRESPONDING FIELDS OF TABLE lt\_aufk  
\*      WHERE ernam IN s\_user  
\*        AND erdat IN s\_datum.  
\*  
\*    LOOP AT lt\_aufk INTO ls\_aufk.  
\*      CLEAR gs\_data.  
\*      MOVE ls\_aufk-aufnr TO gs\_data-object.  
\*      MOVE ls\_aufk-ernam TO gs\_data-user.  
\*      MOVE ls\_aufk-erdat TO gs\_data-date.  
\*      MOVE 'AUFK'        TO gs\_data-tablename.  
\*      MOVE 1 TO gs\_data-counter.  
\*      APPEND gs\_data TO gt\_data.  
\*    ENDLOOP.  
\*  ENDIF.  
  
\*LIKP  
  IF p\_likp EQ 'X'.  
    DATA: lt\_likp TYPE TABLE OF likp.  
    DATA: ls\_likp TYPE likp.  
    REFRESH lt\_likp.  
    SELECT vbeln ernam erdat erzet FROM likp  
      INTO CORRESPONDING FIELDS OF TABLE lt\_likp  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_likp INTO ls\_likp.  
      CLEAR gs\_data.  
      MOVE ls\_likp-vbeln TO gs\_data-object.  
      MOVE ls\_likp-ernam TO gs\_data-user.  
      MOVE ls\_likp-erdat TO gs\_data-date.  
      MOVE ls\_likp-erzet TO gs\_data-time.  
      MOVE 'LIKP'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*VTTK  
  IF p\_vttk EQ 'X'.  
    DATA: lt\_vttk TYPE TABLE OF vttk.  
    DATA: ls\_vttk TYPE vttk.  
    REFRESH lt\_vttk.  
    SELECT tknum ernam erdat erzet FROM vttk  
      INTO CORRESPONDING FIELDS OF TABLE lt\_vttk  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_vttk INTO ls\_vttk.  
      CLEAR gs\_data.  
      MOVE ls\_vttk-tknum TO gs\_data-object.  
      MOVE ls\_vttk-ernam TO gs\_data-user.  
      MOVE ls\_vttk-erdat TO gs\_data-date.  
      MOVE ls\_vttk-erzet TO gs\_data-time.  
      MOVE 'VTTK'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*VBRK  
  IF p\_vbrk EQ 'X'.  
    DATA: lt\_vbrk TYPE TABLE OF vbrk.  
    DATA: ls\_vbrk TYPE vbrk.  
    REFRESH lt\_vbrk.  
    SELECT vbeln ernam erdat erzet FROM vbrk  
      INTO CORRESPONDING FIELDS OF TABLE lt\_vbrk  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_vbrk INTO ls\_vbrk.  
      CLEAR gs\_data.  
      MOVE ls\_vbrk-vbeln TO gs\_data-object.  
      MOVE ls\_vbrk-ernam TO gs\_data-user.  
      MOVE ls\_vbrk-erdat TO gs\_data-date.  
      MOVE ls\_vbrk-erzet TO gs\_data-time.  
      MOVE 'VBRK'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*MAra  
  IF p\_mara EQ 'X'.  
    DATA: lt\_mara TYPE TABLE OF mara.  
    DATA: ls\_mara TYPE mara.  
    REFRESH lt\_mara.  
    SELECT matnr ernam ersda FROM mara  
      INTO CORRESPONDING FIELDS OF TABLE lt\_mara  
      WHERE ernam IN s\_user  
        AND ersda IN s\_datum.  
  
    LOOP AT lt\_mara INTO ls\_mara.  
      CLEAR gs\_data.  
      MOVE ls\_mara-matnr TO gs\_data-object.  
      MOVE ls\_mara-ernam TO gs\_data-user.  
      MOVE ls\_mara-ersda TO gs\_data-date.  
      MOVE 'MARA'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*crhd  
  IF p\_crhd EQ 'X'.  
    DATA: lt\_crhd TYPE TABLE OF crhd.  
    DATA: ls\_crhd TYPE crhd.  
    REFRESH lt\_crhd.  
    SELECT arbpl begda aenam\_grnd werks FROM crhd  
      INTO CORRESPONDING FIELDS OF TABLE lt\_crhd  
      WHERE aenam\_grnd IN s\_user  
        AND begda IN s\_datum.  
  
    LOOP AT lt\_crhd INTO ls\_crhd.  
      CLEAR gs\_data.  
      MOVE ls\_crhd-arbpl TO gs\_data-object.  
      MOVE ls\_crhd-werks TO gs\_data-object2.  
      MOVE ls\_crhd-aenam\_grnd TO gs\_data-user.  
      MOVE ls\_crhd-begda TO gs\_data-date.  
      MOVE 'CRHD'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*STKO  
  IF p\_stko EQ 'X'.  
    DATA: lt\_stko TYPE TABLE OF stko.  
    DATA: ls\_stko TYPE stko.  
    REFRESH lt\_stko.  
    SELECT stlnr annam andat FROM stko  
      INTO CORRESPONDING FIELDS OF TABLE lt\_stko  
      WHERE annam IN s\_user  
        AND andat IN s\_datum.  
  
    LOOP AT lt\_stko INTO ls\_stko.  
      CLEAR gs\_data.  
      MOVE ls\_stko-stlnr TO gs\_data-object.  
      MOVE ls\_stko-annam TO gs\_data-user.  
      MOVE ls\_stko-andat TO gs\_data-date.  
      MOVE 'STKO'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*PLKO  
  IF p\_plko EQ 'X'.  
    DATA: lt\_plko TYPE TABLE OF plko.  
    DATA: ls\_plko TYPE plko.  
    REFRESH lt\_plko.  
    SELECT plnal annam andat FROM plko  
      INTO CORRESPONDING FIELDS OF TABLE lt\_plko  
      WHERE annam IN s\_user  
        AND andat IN s\_datum.  
  
    LOOP AT lt\_plko INTO ls\_plko.  
      CLEAR gs\_data.  
      MOVE ls\_plko-plnal TO gs\_data-object.  
      MOVE ls\_plko-annam TO gs\_data-user.  
      MOVE ls\_plko-andat TO gs\_data-date.  
      MOVE 'PLKO'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*CABN  
  IF p\_cabn EQ 'X'.  
    DATA: lt\_cabn TYPE TABLE OF cabn.  
    DATA: ls\_cabn TYPE cabn.  
    REFRESH lt\_cabn.  
    SELECT atnam aname adatu FROM cabn  
      INTO CORRESPONDING FIELDS OF TABLE lt\_cabn  
      WHERE aname IN s\_user  
        AND adatu IN s\_datum.  
  
    LOOP AT lt\_cabn INTO ls\_cabn.  
      CLEAR gs\_data.  
      MOVE ls\_cabn-atnam TO gs\_data-object.  
      MOVE ls\_cabn-aname TO gs\_data-user.  
      MOVE ls\_cabn-adatu TO gs\_data-date.  
      MOVE 'CABN'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*AFRU  
  IF p\_afru EQ 'X'.  
    DATA: lt\_afru TYPE TABLE OF afru.  
    DATA: ls\_afru TYPE afru.  
    REFRESH lt\_afru.  
    SELECT rueck ernam ersda FROM afru  
      INTO CORRESPONDING FIELDS OF TABLE lt\_afru  
      WHERE ernam IN s\_user  
        AND ersda IN s\_datum.  
  
    LOOP AT lt\_afru INTO ls\_afru.  
      CLEAR gs\_data.  
      MOVE ls\_afru-rueck TO gs\_data-object.  
      MOVE ls\_afru-ernam TO gs\_data-user.  
      MOVE ls\_afru-ersda TO gs\_data-date.  
      MOVE 'AFRU'        TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*MKPF  
  IF p\_mkpf EQ 'X'.  
    DATA: lt\_mkpf TYPE TABLE OF mkpf.  
    DATA: ls\_mkpf TYPE mkpf.  
    REFRESH lt\_mkpf.  
    SELECT mblnr mjahr usnam cpudt cputm FROM mkpf  
      INTO CORRESPONDING FIELDS OF TABLE lt\_mkpf  
      WHERE usnam IN s\_user  
        AND cpudt IN s\_datum.  
  
    LOOP AT lt\_mkpf INTO ls\_mkpf.  
      CLEAR gs\_data.  
      MOVE ls\_mkpf-mblnr TO gs\_data-object.  
      MOVE ls\_mkpf-mjahr TO gs\_data-object2.  
      MOVE ls\_mkpf-usnam TO gs\_data-user.  
      MOVE ls\_mkpf-cpudt TO gs\_data-date.  
      MOVE ls\_mkpf-cputm TO gs\_data-time.  
      MOVE 1 TO gs\_data-counter.  
      MOVE 'MKPF'        TO gs\_data-tablename.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*CAUFV  
  IF p\_caufv EQ 'X' OR p\_caufv2 EQ 'X' OR p\_caufv3 EQ 'X'.  
    DATA: lt\_caufv TYPE TABLE OF caufv.  
    DATA: ls\_caufv TYPE caufv.  
    REFRESH lt\_caufv.  
    SELECT aufnr autyp ernam erdat erfzeit FROM caufv  
      INTO CORRESPONDING FIELDS OF TABLE lt\_caufv  
      WHERE ( autyp EQ '10' OR  
              autyp EQ '20' OR  
              autyp EQ '30' )  
        AND ernam IN s\_user  
        AND erdat IN s\_datum.  
  
\*CAUFV  
    LOOP AT lt\_caufv INTO ls\_caufv WHERE autyp EQ '10'.  
      CLEAR gs\_data.  
      MOVE ls\_caufv-aufnr   TO gs\_data-object.  
      MOVE ls\_caufv-ernam   TO gs\_data-user.  
      MOVE ls\_caufv-erdat   TO gs\_data-date.  
      MOVE ls\_caufv-erfzeit TO gs\_data-time.  
      MOVE 'CAUFV'          TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  
\*CAUFV2 - PM orders  
    LOOP AT lt\_caufv INTO ls\_caufv WHERE autyp EQ '30'.  
      CLEAR gs\_data.  
      MOVE ls\_caufv-aufnr   TO gs\_data-object.  
      MOVE ls\_caufv-ernam   TO gs\_data-user.  
      MOVE ls\_caufv-erdat   TO gs\_data-date.  
      MOVE ls\_caufv-erfzeit TO gs\_data-time.  
      MOVE 'CAUFV2'          TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  
\*CAUFV3  
    LOOP AT lt\_caufv INTO ls\_caufv WHERE autyp EQ '20'.  
      CLEAR gs\_data.  
      MOVE ls\_caufv-aufnr   TO gs\_data-object.  
      MOVE ls\_caufv-ernam   TO gs\_data-user.  
      MOVE ls\_caufv-erdat   TO gs\_data-date.  
      MOVE ls\_caufv-erfzeit TO gs\_data-time.  
      MOVE 'CAUFV3'          TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*EBAN  
  IF p\_eban EQ 'X'.  
    DATA: lt\_eban TYPE TABLE OF eban.  
    DATA: ls\_eban TYPE eban.  
    REFRESH lt\_eban.  
    SELECT banfn ernam erdat FROM eban  
      INTO CORRESPONDING FIELDS OF TABLE lt\_eban  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_eban INTO ls\_eban.  
      CLEAR gs\_data.  
      MOVE ls\_eban-banfn   TO gs\_data-object.  
      MOVE ls\_eban-ernam   TO gs\_data-user.  
      MOVE ls\_eban-erdat   TO gs\_data-date.  
      MOVE 'EBAN'          TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*EKKO  
  IF p\_ekko EQ 'X'.  
    DATA: lt\_ekko TYPE TABLE OF ekko.  
    DATA: ls\_ekko TYPE ekko.  
    REFRESH lt\_ekko.  
    SELECT ebeln ernam aedat  FROM ekko  
      INTO CORRESPONDING FIELDS OF TABLE lt\_ekko  
      WHERE ernam IN s\_user  
        AND aedat IN s\_datum.  
  
    LOOP AT lt\_ekko INTO ls\_ekko.  
      CLEAR gs\_data.  
      MOVE ls\_ekko-ebeln   TO gs\_data-object.  
      MOVE ls\_ekko-ernam   TO gs\_data-user.  
      MOVE ls\_ekko-aedat   TO gs\_data-date.  
      MOVE 'EKKO'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*LAGP  
  IF p\_lagp EQ 'X'.  
    DATA: lt\_lagp TYPE TABLE OF lagp.  
    DATA: ls\_lagp TYPE lagp.  
    REFRESH lt\_lagp.  
    SELECT lgpla lgtyp uname laedt  FROM lagp  
      INTO CORRESPONDING FIELDS OF TABLE lt\_lagp  
      WHERE uname IN s\_user  
        AND laedt IN s\_datum.  
  
    LOOP AT lt\_lagp INTO ls\_lagp.  
      CLEAR gs\_data.  
      MOVE ls\_lagp-lgpla   TO gs\_data-object.  
      MOVE ls\_lagp-lgtyp   TO gs\_data-object2.  
      MOVE ls\_lagp-lgnum   TO gs\_data-object3.  
      MOVE ls\_lagp-uname   TO gs\_data-user.  
      MOVE ls\_lagp-laedt   TO gs\_data-date.  
      MOVE 'LAGP'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*LTAK  
  IF p\_ltak EQ 'X'.  
    DATA: lt\_ltak TYPE TABLE OF ltak.  
    DATA: ls\_ltak TYPE ltak.  
    REFRESH lt\_ltak.  
    SELECT tanum lgnum bname bdatu FROM ltak  
      INTO CORRESPONDING FIELDS OF TABLE lt\_ltak  
      WHERE bname IN s\_user  
        AND bdatu IN s\_datum.  
  
    LOOP AT lt\_ltak INTO ls\_ltak.  
      CLEAR gs\_data.  
      MOVE ls\_ltak-tanum   TO gs\_data-object.  
      MOVE ls\_ltak-lgnum   TO gs\_data-object2.  
      MOVE ls\_ltak-bname   TO gs\_data-user.  
      MOVE ls\_ltak-bdatu   TO gs\_data-date.  
      MOVE 'LTAK'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
\*VEKP  
  IF p\_vekp EQ 'X'.  
    DATA: lt\_vekp TYPE TABLE OF vekp.  
    DATA: ls\_vekp TYPE vekp.  
    REFRESH lt\_vekp.  
    SELECT venum ernam aedat  FROM vekp  
      INTO CORRESPONDING FIELDS OF TABLE lt\_vekp  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_vekp INTO ls\_vekp.  
      CLEAR gs\_data.  
      MOVE ls\_vekp-venum   TO gs\_data-object.  
      MOVE ls\_vekp-ernam   TO gs\_data-user.  
      MOVE ls\_vekp-erdat   TO gs\_data-date.  
      MOVE 'VEKP'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*sdc01 - begin  
  
\*KNA1  
  IF p\_kna1 EQ 'X'.  
    DATA: lt\_kna1 TYPE TABLE OF kna1.  
    DATA: ls\_kna1 TYPE kna1.  
    REFRESH lt\_kna1.  
    SELECT kunnr ernam erdat FROM kna1  
      INTO CORRESPONDING FIELDS OF TABLE lt\_kna1  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_kna1 INTO ls\_kna1.  
      CLEAR gs\_data.  
      MOVE ls\_kna1-kunnr   TO gs\_data-object.  
      MOVE ls\_kna1-ernam   TO gs\_data-user.  
      MOVE ls\_kna1-erdat   TO gs\_data-date.  
      MOVE 'KNA1'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
  
\*LFA1  
  IF p\_lfa1 EQ 'X'.  
    DATA: lt\_lfa1 TYPE TABLE OF lfa1.  
    DATA: ls\_lfa1 TYPE lfa1.  
    REFRESH lt\_lfa1.  
    SELECT lifnr ernam erdat  FROM lfa1  
      INTO CORRESPONDING FIELDS OF TABLE lt\_lfa1  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_lfa1 INTO ls\_lfa1.  
      CLEAR gs\_data.  
      MOVE ls\_lfa1-lifnr   TO gs\_data-object.  
      MOVE ls\_lfa1-ernam   TO gs\_data-user.  
      MOVE ls\_lfa1-erdat   TO gs\_data-date.  
      MOVE 'LFA1'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
  
\*KONH  
  IF p\_konh EQ 'X'.  
    DATA: lt\_konh TYPE TABLE OF konh.  
    DATA: ls\_konh TYPE konh.  
    REFRESH lt\_konh.  
    SELECT knumh ernam erdat  FROM konh  
      INTO CORRESPONDING FIELDS OF TABLE lt\_konh  
      WHERE ernam IN s\_user  
        AND erdat IN s\_datum.  
  
    LOOP AT lt\_konh INTO ls\_konh.  
      CLEAR gs\_data.  
      MOVE ls\_konh-knumh   TO gs\_data-object.  
      MOVE ls\_konh-ernam   TO gs\_data-user.  
      MOVE ls\_konh-erdat   TO gs\_data-date.  
      MOVE 'KONH'         TO gs\_data-tablename.  
      MOVE 1 TO gs\_data-counter.  
      APPEND gs\_data TO gt\_data.  
    ENDLOOP.  
  ENDIF.  
  
  
\*sdc01 - end  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  DATA: lv\_pernr LIKE  gt\_data-user.  
\* Descriptions  
  LOOP AT gt\_data INTO gs\_data.  
    CASE gs\_data-tablename.  
      WHEN 'VBAK'.  
        MOVE 'SD Sales Document' TO gs\_data-object\_desc.  
      WHEN 'VBRK'.  
        MOVE 'SD Billing Document' TO gs\_data-object\_desc.  
      WHEN 'VTTK'.  
        MOVE 'Shipment' TO gs\_data-object\_desc.  
      WHEN 'CAUFV'.  
        MOVE 'PP/CO Order'      TO gs\_data-object\_desc.  
      WHEN 'CAUFV2'.  
        MOVE 'PM Order'      TO gs\_data-object\_desc.  
      WHEN 'CAUFV3'.  
        MOVE 'Network'      TO gs\_data-object\_desc.  
      WHEN 'EBAN'.  
        MOVE 'Purchase Requisition' TO gs\_data-object\_desc.  
      WHEN 'EKKO'.  
        MOVE 'Purchasing Document' TO gs\_data-object\_desc.  
\*      when 'EKET'.  
\*        move 'Scheduling Agreement Sched lin' to gs\_data-object\_desc.  
      WHEN 'LIKP'.  
        MOVE ' SD Document: Delivery' TO gs\_data-object\_desc.  
      WHEN 'MARA'.  
        MOVE 'Material number' TO gs\_data-object\_desc.  
      WHEN 'CRHD'.  
        MOVE 'Work center' TO gs\_data-object\_desc.  
      WHEN 'PLKO'.  
        MOVE 'Routing' TO gs\_data-object\_desc.  
      WHEN 'STKO'.  
        MOVE 'Bill of material' TO gs\_data-object\_desc.  
      WHEN 'AFRU'.  
        MOVE 'Confirmation PP-order' TO gs\_data-object\_desc.  
      WHEN 'MKPF'.  
        MOVE 'Material document' TO gs\_data-object\_desc.  
      WHEN 'CABN'.  
        MOVE 'Characteristic' TO gs\_data-object\_desc.  
      WHEN 'VEKP'.  
        MOVE 'Handling unit' TO gs\_data-object\_desc.  
      WHEN 'LTAK'.  
        MOVE 'WM transfer order header' TO gs\_data-object\_desc.  
      WHEN 'LAGP'.  
        MOVE 'Storage bin' TO gs\_data-object\_desc.  
      WHEN 'KNA1'.  
        MOVE 'Customers' TO gs\_data-object\_desc.  
      WHEN 'LFA1'.  
        MOVE 'Vendors' TO gs\_data-object\_desc.  
      WHEN 'KONH'.  
        MOVE 'Conditions' TO gs\_data-object\_desc.  
      WHEN 'BKPF'.  
        MOVE 'Accounting Documents' TO gs\_data-object\_desc.  
      WHEN 'PRPS'.  
        MOVE 'WBS Elements' TO gs\_data-object\_desc.  
      WHEN 'AUFK'.  
        MOVE 'AUFK orders' TO gs\_data-object\_desc.  
    ENDCASE.  
    CLEAR lv\_pernr.  
    SELECT SINGLE persnumber FROM usr21 INTO lv\_pernr WHERE bname = gs\_data-user.  
    SELECT SINGLE name\_first name\_last FROM adrp INTO CORRESPONDING FIELDS OF gs\_data  
      WHERE persnumber = lv\_pernr.  
    MODIFY gt\_data FROM gs\_data.  
  ENDLOOP.  
  
ENDFORM.                    " get\_data  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Module  status\_9000  OUTPUT  
\*&---------------------------------------------------------------------\*  
MODULE status\_9000 OUTPUT.  
  SET PF-STATUS 'SCREEN\_9000'.  
  SET TITLEBAR 'TIT\_9000'.  
ENDMODULE.                 " status\_9000  OUTPUT  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Module  init\_control\_processing\_9000  OUTPUT  
\*&---------------------------------------------------------------------\*  
MODULE init\_control\_processing\_9000 OUTPUT.  
  CHECK my\_container IS INITIAL.  
  
\* Create container for ALV GRID  
  CREATE OBJECT my\_container  
    EXPORTING  
      side  = cl\_gui\_docking\_container=>dock\_at\_bottom  
\*     extension = 1200  
      ratio = 95  
      repid = g\_repid  
      dynnr = '9000'.  
  
  
  CHECK my\_grid IS INITIAL.  
  CREATE OBJECT my\_grid  
    EXPORTING  
      i\_parent = my\_container.  
  
  PERFORM create\_fieldcatalog.  
  
  CLEAR it\_variant.  
  it\_variant-report = g\_repid.  
  IF NOT p\_vari IS INITIAL.  
    MOVE p\_vari  TO it\_variant-variant.  
    MOVE g\_repid TO it\_variant-report.  
  ENDIF.  
  
\* Sorting  
  DATA: ls\_sort LIKE LINE OF it\_sort.  
  ls\_sort-spos = '1'.  
  ls\_sort-fieldname = 'OBJECT\_DESC'.  
  ls\_sort-up = 'X'.  
  APPEND ls\_sort TO it\_sort.  
  
\* allow to select multiple lines  
  gs\_layout\_alv-sel\_mode = 'A'.  
  
  CALL METHOD my\_grid->set\_table\_for\_first\_display  
    EXPORTING  
      is\_variant      = it\_variant  
      i\_save          = 'A'  
      i\_default       = 'X'  
      is\_layout       = gs\_layout\_alv  
    CHANGING  
      it\_outtab       = gt\_data[]  
      it\_fieldcatalog = it\_fieldcat\_lvc[]  
      it\_sort         = it\_sort.  
  
  
  CREATE OBJECT event\_handler.  
  SET HANDLER event\_handler->hotspot\_click         FOR my\_grid.  
  SET HANDLER event\_handler->handle\_user\_command   FOR my\_grid.  
  SET HANDLER event\_handler->handle\_toolbar        FOR my\_grid.  
  CALL METHOD my\_grid->set\_toolbar\_interactive.  
  
ENDMODULE.                 " init\_control\_processing\_9000  OUTPUT  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Module  user\_command\_9000  INPUT  
\*&---------------------------------------------------------------------\*  
MODULE user\_command\_9000 INPUT.  
  CASE sy-ucomm.  
    WHEN 'CANC' OR 'EXIT' OR 'BACK'.  
      CALL METHOD my\_grid->free.  
      CALL METHOD my\_container->free.  
      FREE: my\_grid, my\_container.  
      LEAVE TO SCREEN 0.  
    WHEN OTHERS.  
      CALL METHOD cl\_gui\_cfw=>dispatch.  
  ENDCASE.  
ENDMODULE.                 " user\_command\_9000  INPUT  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  create\_fieldcatalog  
\*&---------------------------------------------------------------------\*  
FORM create\_fieldcatalog.  
  CALL FUNCTION 'LVC\_FIELDCATALOG\_MERGE'  
    EXPORTING  
      i\_structure\_name   = 'YBC\_USER\_STATISTICS\_ALV'  
      i\_bypassing\_buffer = 'X'  
    CHANGING  
      ct\_fieldcat        = it\_fieldcat\_lvc[].  
  PERFORM change\_fieldcatalog.  
ENDFORM.                    " create\_fieldcatalog  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  change\_fieldcatalog  
\*&---------------------------------------------------------------------\*  
FORM change\_fieldcatalog .  
  LOOP AT it\_fieldcat\_lvc.  
    CASE it\_fieldcat\_lvc-fieldname.  
      WHEN 'TABLENAME'.  
        it\_fieldcat\_lvc-no\_out = 'X'.  
      WHEN 'USER'.  
        it\_fieldcat\_lvc-hotspot = 'X'.  
      WHEN 'OBJECT'.  
        it\_fieldcat\_lvc-hotspot = 'X'.  
    ENDCASE.  
  
    MODIFY it\_fieldcat\_lvc.  
  ENDLOOP.  
  
ENDFORM.                    " change\_fieldcatalog  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  Refresh\_list  
\*&---------------------------------------------------------------------\*  
FORM refresh\_list.  
  PERFORM get\_data.  
  CALL METHOD my\_grid->refresh\_table\_display  
    EXCEPTIONS  
      finished = 1  
      OTHERS   = 2.  
ENDFORM.                    "Refresh\_list  
  
\*}   INSERT