Products
Services & Support
About SCN
Downloads
Industries
Training & Education
Partnership
Developer Center
Lines of Business
University Alliances
Events & Webinars
Innovation

ABAP Development / ABAP Development and Programming

Tree ALV

Added by Ritesh Jha, last edited by Smruti Ranjan Mohanty on Oct 23, 2013

```
*&amp;---------------------------------------------------------------------*
*&amp; Report  ZER14_AFUDC1
*&amp;
*&amp;---------------------------------------------------------------------*
*&amp;
*&amp;
*&amp;---------------------------------------------------------------------*
REPORT  ZER14_AFUDC1 MESSAGE-ID ZER14.
*&amp;---------------------------------------------------------------------*
*&amp; DECLERATION OF TYPE POOL.
*&amp;
*&amp;---------------------------------------------------------------------*
TYPE-POOLS SLIS.
*&amp;---------------------------------------------------------------------*
*&amp; DECLERATION OF TABLES.
*&amp;
*&amp;---------------------------------------------------------------------*
TABLES: AUFK,BKPF,PROJ,TJ02T,PRPS,JCDS,COEP.
*&amp;---------------------------------------------------------------------*
*          TYPE DECLARATION
*&amp;---------------------------------------------------------------------*
TYPES:
*FOR aufk TABLE
     BEGIN OF T_AUFK,
       AUFNR TYPE AUFK-AUFNR,  "ORDER NUMBER
       OBJNR TYPE AUFK-OBJNR,   "OBJECT NUMBER
       PSPEL TYPE AUFK-PSPEL,   "WBS ELEMENT
       STAT TYPE JCDS-STAT,     "OBJECT STATUS
       UDATE TYPE JCDS-UDATE,   "CREATION DATE
       INACT TYPE JCDS-INACT,   "INACTIVE STATUS
       UTIME TYPE JCDS-UTIME,   "TIME CHANGED
       CHGNR TYPE JCDS-CHGNR,   "Change number
       END OF T_AUFK,
*FOR PROJ TABLE
```

```
   BEGIN OF T_PROJ,
     PSPNR TYPE PROJ-PSPNR,  "PROJECT DEFINATION (INITIAL)
     PSPID TYPE PROJ-PSPID,  "PROJECT DEFINATION
     POST1 TYPE PROJ-POST1,  "PROJECT DESCRIPTION
     PROFL TYPE PROJ-PROFL,
     END OF T_PROJ,
*FOR PRPS TABLE
     BEGIN OF T_PRPS,
       PSPNR TYPE PRPS-PSPNR,  "WBS ELEMENT
       OBJNR TYPE PRPS-OBJNR,  "OBJECT NUMBER
       PSPHI TYPE PRPS-PSPHI,  "CURRENT NUMBER OF THE APPROPIATE OBJECT
       POST1 TYPE PRPS-POST1,  "WBS DESCRIPTION
       END OF T_PRPS,
*FOR COEP TABLE
       BEGIN OF T_COEP,
         WTGBTR TYPE COEP-WTGBTR,  "Total Value in Transaction Currency
         OBJNR TYPE COEP-OBJNR,    "OBJECT NUMBER
         END OF T_COEP,
*FOR COEP TABLE
       BEGIN OF T_COEP1,
         WTGBTR TYPE COEP-WTGBTR,  "Total Value in Transaction Currency
         OBJNR TYPE COEP-OBJNR,    "OBJECT NUMBER
         END OF T_COEP1,
*FOR TJ02T TABLE
         BEGIN OF T_TJ02T,
           ISTAT TYPE TJ02T-ISTAT,  "System status
           TXT04 TYPE TJ02T-TXT04,  "ndividual status of an object (short form)
           END OF T_TJ02T,
* FOR T056P TABLE
           BEGIN OF T_T056P,
             ZSOLL TYPE T056P-ZSOLL, "Interest Rate
             DATAB TYPE T056P-DATAB, "Effective-From Date
             END OF T_T056P,
* FOR BKPF TABLE
             BEGIN OF T_BKPF,
               GJAHR TYPE BKPF,
               END OF T_BKPF,
*FOR FINAL INTERNAL TABLE
             BEGIN OF T_FINAL,
               AUFNR TYPE AUFK-AUFNR,  "ORDER NUMBER
               OBJNR TYPE AUFK-OBJNR,  "OBJECT NUMBER
               PSPEL TYPE AUFK-PSPEL,  "WBS ELEMENT
               POST1 TYPE PROJ-POST1,  "PROJECT DESCRIPTION
               POST11 TYPE PRPS-POST1, "WBS DESCRIPTION
               STAT TYPE JCDS-STAT,    "OBJECT STATUS
               UDATE TYPE JCDS-UDATE,  "CREATION DATE
               INACT TYPE JCDS-INACT,  "INACTIVE STATUS
               UTIME TYPE JCDS-UTIME,  "TIME CHANGED
               PSPNR TYPE PRPS-PSPNR,  "WBS ELEMENT
               PSPHI TYPE PRPS-PSPHI,  "CURRENT NUMBER OF THE APPROPIATE OBJECT
               WTGBTR TYPE COEP-WTGBTR, "Total Value in Transaction Currency
               WTGBTR1 TYPE COEP-WTGBTR, "Total Value in Transaction Currency
               ZSOLL TYPE T056P-ZSOLL,   "Interest Rate
               ZSOLL1 TYPE T056P-ZSOLL,  "Interest Rate
               V_VAR1 TYPE P DECIMALS 2,  "FOR CALCULATION OF DEBT AMOUNT
               V_VAR2 TYPE P DECIMALS 2,  "FOR CLCULATION OF EQUITY AMOUNT
               V_TOTAL TYPE P DECIMALS 2, " TOTAL OF V_VAR1 AND V_VAR2
              V_E_BAL TYPE P DECIMALS 2,  "ENDING TOTAL
               END OF T_FINAL.
*&amp;---------------------------------------------------------------------*
*          PARAMETERS FOR SELECTION-SCREEN
*&amp;---------------------------------------------------------------------*
*          SELECTION SCREEN BLOCK FOR SELECTING VALUE
*&amp;---------------------------------------------------------------------*
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.
SELECT-OPTIONS :
                   S_PROFL FOR PROJ-PROFL,
                 S_PSPNR FOR PROJ-PSPNR.
PARAMETERS :
*              P_MONAT TYPE BKPF-MONAT OBLIGATORY,
              P_GJAHR TYPE BKPF-GJAHR OBLIGATORY,
              V_FM_DT TYPE SY-DATUM OBLIGATORY,
              V_TO_DT TYPE SY-DATUM OBLIGATORY.
SELECTION-SCREEN END OF BLOCK B1.
*&amp;---------------------------------------------------------------------*
*          VARIABLE DECLARATIONS
*&amp;---------------------------------------------------------------------*
DATA: V_ISTAT TYPE TJ02T-ISTAT,
       V_CAL TYPE  P DECIMALS 2,  " FOR CALCULATION OF wa_final-zsoll / 12
       V_CAL1 TYPE P DECIMALS 2.  " FOR CALCULATION OF wa_final-zsoll1 / 12
*&amp;---------------------------------------------------------------------*
*          CONTANTS DECLARATION
*&amp;---------------------------------------------------------------------*
CONSTANTS : C_X(1) VALUE 'X',       " value used to set X for a field
            C_REPID(40) VALUE SY-REPID,        " report id
            C_TECO(4) VALUE 'TECO', " object status description
            C_CHECK(1) VALUE 'X'.  " value used to set X for a field
CLASS CL_GUI_COLUMN_TREE DEFINITION LOAD.
CLASS CL_GUI_CFW DEFINITION LOAD.
DATA TREE1  TYPE REF TO CL_GUI_ALV_TREE.
DATA MR_TOOLBAR TYPE REF TO CL_GUI_TOOLBAR.
INCLUDE <ICON>.
INCLUDE BCALV_TOOLBAR_EVENT_RECEIVER.
INCLUDE BCALV_TREE_EVENT_RECEIVER.
DATA: TOOLBAR_EVENT_RECEIVER TYPE REF TO LCL_TOOLBAR_EVENT_RECEIVER.
DATA: IT_FIELD TYPE LVC_T_FCAT, "Fieldcatalog
      OK_CODE LIKE SY-UCOMM.           "OK-Code
*------------------------------------
*          INTERNAL TABLE
*------------------------------------
DATA: IT_AUFK TYPE STANDARD TABLE OF T_AUFK, " INTERNAL TABLE OF TYPE I_AUFK
      IT_PROJ TYPE STANDARD TABLE OF T_PROJ, " INTERNAL TABLE OF TYPE I_PROJ
      IT_PRPS TYPE STANDARD TABLE OF T_PRPS, " INTERNAL TABLE OF TYPE I_PRPS
      IT_COEP TYPE STANDARD TABLE OF T_COEP, " INTERNAL TABLE OF TYPE I_COEP
      IT_COEP1 TYPE STANDARD TABLE OF T_COEP1, " INTERNAL TABLE OF TYPE I_COPE1
      IT_TJ02T TYPE STANDARD TABLE OF T_TJ02T, " INTERNAL TABLE OF TYPE I_TJ02P
      IT_T056P TYPE STANDARD TABLE OF T_T056P, " INTERNAL TABLE OF TYPE I_T056T
      IT_FINAL TYPE STANDARD TABLE OF T_FINAL, " INTERNAL TABLE OF TYPE I_FIANL
      IT_BKPF  TYPE STANDARD TABLE OF T_BKPF,
      IT_FINAL1 TYPE STANDARD TABLE OF T_FINAL.
*------------------------------------
*          WORK AREA
*------------------------------------
DATA:WA_AUFK TYPE T_AUFK,
     WA_PROJ TYPE T_PROJ,
     WA_PRPS TYPE T_PRPS,
     WA_COEP TYPE T_COEP,
     WA_COEP1 TYPE T_COEP1,
     WA_TJ02T TYPE T_TJ02T,
     WA_T056P TYPE T_T056P,
     WA_FINAL TYPE T_FINAL,
     WA_BKPF  TYPE T_BKPF,
       WA_FINAL1 TYPE T_FINAL.
START-OF-SELECTION.
*********************************************************
*  perform for passing the hard code value
*********************************************************
   *  PERFORM HARD_CODE.
******************************************************************
*             PERFORM for get the value
******************************************************************
  PERFORM GET_DATA.
END-OF-SELECTION.
  CALL SCREEN 9000.
*&amp;---------------------------------------------------------------------*
*&amp;      Module  STATUS_9000  OUTPUT
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE STATUS_9000 OUTPUT.
  SET PF-STATUS 'ZMAIN'.
*  SET TITLEBAR 'xxx'.
  IF TREE1 IS INITIAL.
    PERFORM INIT_TREE.
  ENDIF.
  CALL METHOD CL_GUI_CFW=>FLUSH.
ENDMODULE.                 " STATUS_9000  OUTPUT
*&---------------------------------------------------------------------*
*&       Form  INIT_TREE
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM INIT_TREE .
  IT_FINAL1[] = IT_FINAL[] .
*   create fieldcatalog for structure sflight
  PERFORM  BUILD_FIELDCATALOG.
```

```
* create container for alv-tree
  DATA: ALV_CONTAINOR(30) TYPE C,
        L_CUSTOM_CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER.
  ALV_CONTAINOR = 'TREE1'.
  IF SY-BATCH IS INITIAL.
    CREATE OBJECT L_CUSTOM_CONTAINER
      EXPORTING
        CONTAINER_NAME              = 'ALV_CONTAINOR'
      EXCEPTIONS
        CNTL_ERROR                  = 1
        CNTL_SYSTEM_ERROR           = 2
        CREATE_ERROR                = 3
        LIFETIME_ERROR              = 4
        LIFETIME_DYNPRO_DYNPRO_LINK = 5.
    IF SY-SUBRC <> 0.
      MESSAGE X208(00) WITH 'ERROR'.                        "#EC NOTEXT
    ENDIF.
  ENDIF.
* create tree control
  CREATE OBJECT TREE1
    EXPORTING
      PARENT                      = L_CUSTOM_CONTAINER
      NODE_SELECTION_MODE         = CL_GUI_COLUMN_TREE=>NODE_SEL_MODE_MULTIPLE
      ITEM_SELECTION              = SPACE
      NO_HTML_HEADER              = ''
      NO_TOOLBAR                  = ''
    EXCEPTIONS
      CNTL_ERROR                  = 1
      CNTL_SYSTEM_ERROR           = 2
      CREATE_ERROR                = 3
      LIFETIME_ERROR              = 4
      ILLEGAL_NODE_SELECTION_MODE = 5
      FAILED                      = 6
      ILLEGAL_COLUMN_NAME         = 7.
  IF SY-SUBRC <> 0.
    MESSAGE X208(00) WITH 'ERROR'.                          "#EC NOTEXT
  ENDIF.
* create Hierarchy-header
  DATA L_HIERARCHY_HEADER TYPE TREEV_HHDR.
  PERFORM BUILD_HIERARCHY_HEADER CHANGING L_HIERARCHY_HEADER.
* create info-table for html-header
  DATA: LT_LIST_COMMENTARY TYPE SLIS_T_LISTHEADER,
        L_LOGO             TYPE SDYDO_VALUE.
*  perform build_comment using
*                 lt_list_commentary
*                 l_logo.
* repid for saving variants
  DATA: LS_VARIANT TYPE DISVARIANT.
  LS_VARIANT-REPORT = SY-REPID.
  BREAK SPCANDXX.
  REFRESH IT_FINAL.
* create emty tree-control
  CALL METHOD TREE1->SET_TABLE_FOR_FIRST_DISPLAY
```

```
    EXPORTING
      IS_HIERARCHY_HEADER = L_HIERARCHY_HEADER
      IT_LIST_COMMENTARY  = LT_LIST_COMMENTARY
      I_LOGO              = L_LOGO
      I_BACKGROUND_ID     = 'ALV_BACKGROUND'
      I_SAVE              = 'A'
      IS_VARIANT          = LS_VARIANT
    CHANGING
      IT_OUTTAB           = IT_FINAL[] "table must be emty !!
      IT_FIELDCATALOG     = IT_FIELD[].
  BREAK SPCANDXX.
* create hierarchy
  PERFORM CREATE_HIERARCHY.
* add own functioncodes to the toolbar
  PERFORM CHANGE_TOOLBAR.
* register events
*  perform register_events.
* adjust column_width
* call method tree1->COLUMN_OPTIMIZE.
ENDFORM.                    " INIT_TREE
*&---------------------------------------------------------------------*
*&      Form  build_fieldcatalog
*&---------------------------------------------------------------------*
*
*----------------------------------------------------------------------*
FORM BUILD_FIELDCATALOG .
  DATA: WA_FIELD TYPE LVC_S_FCAT.
*  get fieldcatalog
  CALL FUNCTION 'LVC_FIELDCATALOG_MERGE'
    EXPORTING
      I_STRUCTURE_NAME = 'ZSTRUC'
    CHANGING
      CT_FIELDCAT      = IT_FIELD.
  SORT IT_FIELD BY SCRTEXT_L.
* change fieldcatalog
*  data: wa_field type lvc_s_fcat.
  LOOP AT IT_FIELD INTO WA_FIELD.
    CASE WA_FIELD-FIELDNAME.
      WHEN 'POST1' OR 'POST11' OR 'AUFNR'.
        WA_FIELD-NO_OUT = 'X'.
        WA_FIELD-KEY    = ''.
      WHEN 'WTGBTR' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-003. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'WTGBTR1' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-004. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'ZSOLL' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-005. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'ZSOLL1'.
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-006. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'V_VAR1' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-007. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'V_VAR2' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-008. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'V_TOTAL' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-009. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
      WHEN 'V_E_BAL' .
        WA_FIELD-OUTPUTLEN = 20.       " output length on screen
        WA_FIELD-COLTEXT = TEXT-010. " header information
        WA_FIELD-DO_SUM = 'X'.       " DOING SUM
    ENDCASE.
    MODIFY IT_FIELD FROM WA_FIELD.
  ENDLOOP.
ENDFORM..                    "build_fieldcatalog
*&amp;---------------------------------------------------------------------*
*&amp;      Form  BUILD_HIERARCHY_HEADER
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      <--P_L_HIERARCHY_HEADER  text
*----------------------------------------------------------------------*
FORM BUILD_HIERARCHY_HEADER  CHANGING P_HIERARCHY_HEADER TYPE TREEV_HHDR.
  P_HIERARCHY_HEADER-HEADING = 'Business Segment'.          "#EC NOTEXT
  P_HIERARCHY_HEADER-TOOLTIP =
                         'This is the Hierarchy Header !'.  "#EC NOTEXT
  P_HIERARCHY_HEADER-WIDTH = 30.
  P_HIERARCHY_HEADER-WIDTH_PIX = ''.
ENDFORM.                    " BUILD_HIERARCHY_HEADER
*&---------------------------------------------------------------------*
*&      Form  BUILD_COMMENT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_LT_LIST_COMMENTARY  text
*      -->P_L_LOGO  text
*----------------------------------------------------------------------*
FORM BUILD_COMMENT  USING
 PT_LIST_COMMENTARY TYPE SLIS_T_LISTHEADER
      P_LOGO             TYPE SDYDO_VALUE.
  DATA: LS_LINE TYPE SLIS_LISTHEADER.
*
* LIST HEADING LINE: TYPE H
  CLEAR LS_LINE.
  LS_LINE-TYP  = 'H'.
* LS_LINE-KEY:  NOT USED FOR THIS TYPE
  LS_LINE-INFO = 'ALV-tree-demo: flight-overview'.          "#EC NOTEXT
  APPEND LS_LINE TO PT_LIST_COMMENTARY.
* STATUS LINE: TYPE S
  CLEAR LS_LINE.
  LS_LINE-TYP  = 'S'.
  LS_LINE-KEY  = 'valid until'.                             "#EC NOTEXT
  LS_LINE-INFO = 'January 29 1999'.                         "#EC NOTEXT
  APPEND LS_LINE TO PT_LIST_COMMENTARY.
  LS_LINE-KEY  = 'time'.
  LS_LINE-INFO = '2.00 pm'.                                 "#EC NOTEXT
  APPEND LS_LINE TO PT_LIST_COMMENTARY.
* ACTION LINE: TYPE A
  CLEAR LS_LINE.
  LS_LINE-TYP  = 'A'.
* LS_LINE-KEY:  NOT USED FOR THIS TYPE
  LS_LINE-INFO = 'actual data'.                             "#EC NOTEXT
  APPEND LS_LINE TO PT_LIST_COMMENTARY.
  P_LOGO = 'ENJOYSAP_LOGO'.
ENDFORM.                    " BUILD_COMMENT
*&----------------------------------------------------------------------*
*&      Form  CREATE_HIERARCHY
*&----------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM CREATE_HIERARCHY .
* add data to tree
  DATA: L_PSPNR_KEY TYPE LVC_NKEY,
        L_PSPNR1_KEY TYPE LVC_NKEY,
        L_LAST_KEY TYPE LVC_NKEY.
  SORT IT_FINAL1 BY POST1 POST11 AUFNR.
  LOOP AT IT_FINAL1 INTO WA_FINAL1.
    ON CHANGE OF WA_FINAL1-POST1.
      PERFORM ADD_PSPNR_LINE USING      WA_FINAL1
                                       ''
                              CHANGING L_PSPNR_KEY.
    ENDON.
    ON CHANGE OF WA_FINAL1-POST11.
      PERFORM ADD_PSPNR1_LINE USING     WA_FINAL1
                                       L_PSPNR_KEY
                              CHANGING L_PSPNR1_KEY.
    ENDON.
    PERFORM ADD_COMPLETE_LINE USING     WA_FINAL1
                                         L_PSPNR1_KEY
                                CHANGING L_LAST_KEY.
    CLEAR WA_FINAL1.
  ENDLOOP.
* calculate totals
  CALL METHOD TREE1->UPDATE_CALCULATIONS.
```

```
* this method must be called to send the data to the frontend
  CALL METHOD TREE1->FRONTEND_UPDATE.
ENDFORM.                    " CREATE_HIERARCHY
*&amp;---------------------------------------------------------------------*
*&amp;      Form  GET_DATA
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM GET_DATA .
*&amp;---------------------------------------------------------------------*
*   QUERY TO select ISTAT FROM TJ02T INTO   IT_TJ02T
*&amp;---------------------------------------------------------------------*
  SELECT SINGLE
    ISTAT
    FROM TJ02T
    INTO V_ISTAT
    WHERE
    SPRAS = SY-LANGU AND TXT04 = C_TECO.
  IF SY-SUBRC <> 0.
    MESSAGE E000.
  ENDIF.
*&amp;---------------------------------------------------------------------*
*   QUERY TO select PSPNR PSPID  FROM PROJ INTO IT_PROJ.
*&amp;---------------------------------------------------------------------*
  SELECT
    PSPNR
*    PSPID
    POST1
    FROM PROJ
    INTO TABLE IT_PROJ
    WHERE
    PSPNR IN S_PSPNR
    AND PROFL IN S_PROFL.
  IF SY-SUBRC <> 0.
    MESSAGE E001.
  ENDIF.
*&amp;---------------------------------------------------------------------*
*   QUERY TO select PSPNR OBJNR PSPHI  FROM PRPS INTO IT_PRPS.
*&amp;---------------------------------------------------------------------*
  SELECT
    PSPNR
    OBJNR
    PSPHI
*    PSPID
    POST1
    FROM PRPS
    INTO TABLE IT_PRPS
    FOR ALL ENTRIES IN IT_PROJ
    WHERE PSPHI = IT_PROJ-PSPNR.
  IF SY-SUBRC <> 0 .
```

```
    MESSAGE E002.
  ENDIF.
*&amp;---------------------------------------------------------------------*
*   QUERY TO select AUFNR OBJNR STAT UDAT INACT UTIME FROM AUFK,JCDS INTO IT_AUFK.
*&amp;---------------------------------------------------------------------*
  SELECT
    A~AUFNR
    A~OBJNR
    A~PSPEL
    B~STAT
    B~UDATE
    B~INACT
    B~UTIME
    B~CHGNR
    FROM
    AUFK AS A INNER JOIN JCDS AS B
    ON A~OBJNR = B~OBJNR
    INTO TABLE IT_AUFK
    FOR ALL ENTRIES IN IT_PRPS
    WHERE A~PSPEL = IT_PRPS-PSPNR  AND
          B~STAT = V_ISTAT AND
          B~UDATE BETWEEN V_FM_DT AND V_TO_DT
          AND B~INACT <> C_X.
  IF SY-SUBRC <> 0 .
    MESSAGE E003.
  ENDIF.
*&amp;---------------------------------------------------------------------*
*   QUERY TO select WTGBTR FROM COEP INTO IT_COEP.
*&amp;---------------------------------------------------------------------*
  SELECT
    WTGBTR
    FROM COEP
    INTO TABLE IT_COEP
    FOR ALL ENTRIES IN IT_PRPS
    WHERE
        GJAHR = P_GJAHR  AND OBJNR = IT_PRPS-OBJNR.
  IF SY-SUBRC <> 0 .
    MESSAGE E004.
  ENDIF.
*&amp;---------------------------------------------------------------------*
*   QUERY TO select WTGBTR FROM COEP INTO IT_COEP.
*&amp;---------------------------------------------------------------------*
  SELECT
    WTGBTR
    OBJNR
    FROM COEP
    INTO TABLE IT_COEP1
    FOR ALL ENTRIES IN IT_PRPS
    WHERE
   GJAHR = P_GJAHR AND OBJNR = IT_PRPS-OBJNR.
  IF SY-SUBRC <> 0 .
    MESSAGE E005.
  ENDIF.
```

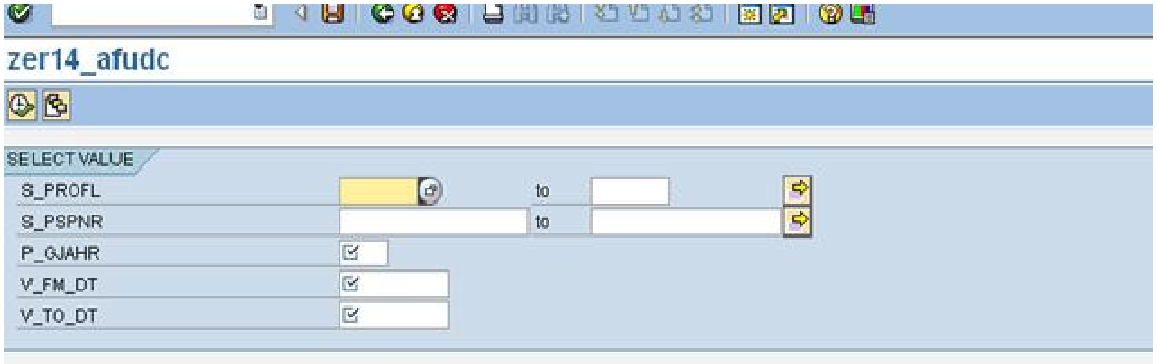
```
*&amp;---------------------------------------------------------------------*
*   QUERY TO select ZSOLL FROM T056P INTO IT_T056P.
*&amp;---------------------------------------------------------------------*
  SELECT
     ZSOLL
     FROM T056P
    INTO TABLE IT_T056P
    WHERE REFERENZ = 'ZNWN DEBT' .
  IF SY-SUBRC <> 0 .
    MESSAGE E006.
  ENDIF.
*&amp;---------------------------------------------------------------------*
*   QUERY TO select ZSOLL FROM T056P INTO IT_T056P.
*&amp;---------------------------------------------------------------------*
  SELECT
     ZSOLL
     FROM T056P
    INTO TABLE IT_T056P
    WHERE REFERENZ = 'ZNWNEQUITY' .
  IF SY-SUBRC <> 0 .
    MESSAGE E007.
  ENDIF.
  BREAK SPCANDXX.
  LOOP AT IT_PRPS INTO WA_PRPS.
    WA_FINAL-PSPNR = WA_PRPS-PSPNR.
    WA_FINAL-OBJNR = WA_PRPS-OBJNR.
    WA_FINAL-PSPHI = WA_PRPS-PSPHI.
    WA_FINAL-POST11 = WA_PRPS-POST1.
    READ TABLE IT_PROJ INTO WA_PROJ WITH KEY PSPNR = WA_PRPS-PSPNR.
    IF SY-SUBRC = 0.
      WA_FINAL-POST1 = WA_PROJ-POST1.
    ELSE.
      MESSAGE E003.
    ENDIF.
    READ TABLE IT_AUFK INTO WA_AUFK  WITH KEY PSPEL = WA_PRPS-PSPNR.
    IF SY-SUBRC = 0.
      WA_FINAL-AUFNR = WA_AUFK-AUFNR.
      WA_FINAL-UDATE = WA_AUFK-UDATE.
      WA_FINAL-INACT = WA_AUFK-INACT.
      WA_FINAL-UTIME = WA_AUFK-UTIME.
    ELSE.
      MESSAGE E003.
    ENDIF.
    READ TABLE IT_COEP INTO WA_COEP WITH KEY OBJNR = WA_PRPS-OBJNR.
    IF SY-SUBRC = 0.
      WA_FINAL-WTGBTR = WA_COEP-WTGBTR.
    ELSE.
      MESSAGE E003.
    ENDIF.
    READ TABLE IT_COEP1 INTO WA_COEP1 WITH KEY OBJNR = WA_PRPS-OBJNR.
    IF SY-SUBRC = 0.
      WA_FINAL-WTGBTR1 = WA_COEP1-WTGBTR.
    ELSE.
```

```
      MESSAGE E003.
    ENDIF.
    WA_FINAL-ZSOLL = WA_T056P-ZSOLL.
    APPEND WA_FINAL TO IT_FINAL.                               "append wa_final into it_final
    LOOP AT IT_FINAL INTO WA_FINAL.
      V_CAL = WA_FINAL-ZSOLL / 12 .
      V_CAL1 = WA_FINAL-ZSOLL1 / 12 .
      WA_FINAL-V_VAR1 =   ( WA_FINAL-WTGBTR  (  V_CAL )  + ( WA_FINAL-WTGBTR1 )  ( V_CAL  / 2
      WA_FINAL-V_VAR2 =   ( WA_FINAL-WTGBTR  ( V_CAL1 )  + ( WA_FINAL-WTGBTR1 )  (  V_CAL1 ) /
      WA_FINAL-V_TOTAL =     ( WA_FINAL-V_VAR1 + WA_FINAL-V_VAR2 ).
      WA_FINAL-V_E_BAL  =   ( WA_FINAL-WTGBTR  WA_FINAL-WTGBTR1  WA_FINAL-V_TOTAL ).
      MODIFY IT_FINAL FROM WA_FINAL INDEX SY-TABIX.
    ENDLOOP.
  ENDLOOP.
ENDFORM.                    " GET_DATA
*&amp;---------------------------------------------------------------------*
*&amp;      Form  CHANGE_TOOLBAR
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
FORM CHANGE_TOOLBAR .
*   * get toolbar control
  CALL METHOD TREE1->GET_TOOLBAR_OBJECT
    IMPORTING
      ER_TOOLBAR = MR_TOOLBAR.
  CHECK NOT MR_TOOLBAR IS INITIAL.
* add seperator to toolbar
  CALL METHOD MR_TOOLBAR->ADD_BUTTON
    EXPORTING
      FCODE     = ''
      ICON      = ''
      BUTN_TYPE = CNTB_BTYPE_SEP
      TEXT      = ''
      QUICKINFO = 'This is a Seperator'.                    "#EC NOTEXT
* add Standard Button to toolbar (for Delete Subtree)
  CALL METHOD MR_TOOLBAR->ADD_BUTTON
    EXPORTING
      FCODE     = 'DELETE'
      ICON      = ' (18)'
      BUTN_TYPE = CNTB_BTYPE_BUTTON
      TEXT      = ''
      QUICKINFO = 'Delete subtree'.                         "#EC NOTEXT
* add Dropdown Button to toolbar (for Insert Line)
  CALL METHOD MR_TOOLBAR->ADD_BUTTON
    EXPORTING
      FCODE     = 'INSERT_LC'
      ICON      = ' (17)'
      BUTN_TYPE = CNTB_BTYPE_DROPDOWN
      TEXT      = ''
      QUICKINFO = 'Insert Line'.                            "#EC NOTEXT
```

```
* set event-handler for toolbar-control
  CREATE OBJECT TOOLBAR_EVENT_RECEIVER.
  SET HANDLER TOOLBAR_EVENT_RECEIVER->ON_FUNCTION_SELECTED
                                                      FOR MR_TOOLBAR.
  SET HANDLER TOOLBAR_EVENT_RECEIVER->ON_TOOLBAR_DROPDOWN
                                                      FOR MR_TOOLBAR.
ENDFORM.                    " CHANGE_TOOLBAR
*&amp;---------------------------------------------------------------------*
*&amp;      Form  ADD_PSPNR_LINE
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_WA_FINAL  text
*      -->P_0958   text
*      <--P_L_PSPNR_KEY  text
*----------------------------------------------------------------------*
FORM ADD_COMPLETE_LINE USING   WA_FINAL1 TYPE T_FINAL
                               P_RELAT_KEY TYPE LVC_NKEY
                     CHANGING  P_NODE_KEY TYPE LVC_NKEY.
  DATA: L_NODE_TEXT TYPE LVC_VALUE. ",
*        wa_final2 LIKE it_final.
* set item-layout
  DATA: LT_ITEM_LAYOUT TYPE LVC_T_LAYI,
        LS_ITEM_LAYOUT TYPE LVC_S_LAYI.
  LS_ITEM_LAYOUT-T_IMAGE = ' (3P)'.
  LS_ITEM_LAYOUT-FIELDNAME = TREE1->C_HIERARCHY_COLUMN_NAME.
  LS_ITEM_LAYOUT-STYLE   =
                        CL_GUI_COLUMN_TREE=>STYLE_INTENSIFD_CRITICAL.
  APPEND LS_ITEM_LAYOUT TO LT_ITEM_LAYOUT.
* add node
  L_NODE_TEXT =  WA_FINAL1-AUFNR.
  DATA: LS_NODE TYPE LVC_S_LAYN.
  LS_NODE-N_IMAGE   = SPACE.
  LS_NODE-EXP_IMAGE = SPACE.
  CALL METHOD TREE1->ADD_NODE
    EXPORTING
      I_RELAT_NODE_KEY = P_RELAT_KEY
      I_RELATIONSHIP   = CL_GUI_COLUMN_TREE=>RELAT_LAST_CHILD
      I_NODE_TEXT      = L_NODE_TEXT
      IS_OUTTAB_LINE   = WA_FINAL1
      IS_NODE_LAYOUT   = LS_NODE
      IT_ITEM_LAYOUT   = LT_ITEM_LAYOUT
    IMPORTING
      E_NEW_NODE_KEY   = P_NODE_KEY.
ENDFORM.                               " add_carrid_line
*&amp;---------------------------------------------------------------------*
*&amp;      Form  ADD_PSPNR1_LINE
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_WA_FINAL  text
*      -->P_L_PSPNR_KEY  text
*      <--P_L_PSPNR1_KEY  text
```

```
*----------------------------------------------------------------------*
FORM ADD_PSPNR1_LINE  USING
                              WA_FINAL1 TYPE T_FINAL
                               P_RELAT_KEY TYPE LVC_NKEY
                     CHANGING  P_NODE_KEY TYPE LVC_NKEY.
  DATA: L_NODE_TEXT TYPE LVC_VALUE.   ",
*        wa_final2 type t_final.
* set item-layout
  DATA: LT_ITEM_LAYOUT TYPE LVC_T_LAYI,
        LS_ITEM_LAYOUT TYPE LVC_S_LAYI.
  LS_ITEM_LAYOUT-T_IMAGE = ' (3Y)'.
  LS_ITEM_LAYOUT-STYLE   =
                        CL_GUI_COLUMN_TREE=>STYLE_INTENSIFIED.
  LS_ITEM_LAYOUT-FIELDNAME = TREE1->C_HIERARCHY_COLUMN_NAME.
  APPEND LS_ITEM_LAYOUT TO LT_ITEM_LAYOUT.
* add node
  L_NODE_TEXT =  WA_FINAL1-POST11.
  DATA: RELAT TYPE INT4.
  RELAT = CL_GUI_COLUMN_TREE=>RELAT_LAST_CHILD.
  CALL METHOD TREE1->ADD_NODE
    EXPORTING
      I_RELAT_NODE_KEY = P_RELAT_KEY
      I_RELATIONSHIP   = RELAT
      I_NODE_TEXT      = L_NODE_TEXT
      IS_OUTTAB_LINE   = WA_FINAL1
      IT_ITEM_LAYOUT   = LT_ITEM_LAYOUT
    IMPORTING
      E_NEW_NODE_KEY   = P_NODE_KEY.
ENDFORM.                    " ADD_PSPNR1_LINE
*&amp;---------------------------------------------------------------------*
*&amp;      Form  ADD_PSPNR_LINE
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*      -->P_WA_FINAL  text
*      -->P_0958   text
*      <--P_L_PSPNR_KEY  text
*----------------------------------------------------------------------*
FORM ADD_PSPNR_LINE USING     WA_FINAL1 TYPE T_FINAL
                               P_RELAT_KEY TYPE LVC_NKEY
                     CHANGING  P_NODE_KEY TYPE LVC_NKEY.
  DATA: L_NODE_TEXT TYPE LVC_VALUE.  " ,
*        wa_final2 TYPE t_final.
* set item-layout
  DATA: LT_ITEM_LAYOUT TYPE LVC_T_LAYI,
        LS_ITEM_LAYOUT TYPE LVC_S_LAYI.
  LS_ITEM_LAYOUT-T_IMAGE = ' (3P)'.
  LS_ITEM_LAYOUT-FIELDNAME = TREE1->C_HIERARCHY_COLUMN_NAME.
  LS_ITEM_LAYOUT-STYLE   =
                        CL_GUI_COLUMN_TREE=>STYLE_INTENSIFD_CRITICAL.
  APPEND LS_ITEM_LAYOUT TO LT_ITEM_LAYOUT.
* add node
  L_NODE_TEXT = WA_FINAL1-POST1.
  DATA: LS_NODE TYPE LVC_S_LAYN.
  LS_NODE-N_IMAGE   = SPACE.
  LS_NODE-EXP_IMAGE = SPACE.
  CALL METHOD TREE1->ADD_NODE
    EXPORTING
      I_RELAT_NODE_KEY = P_RELAT_KEY
      I_RELATIONSHIP   = CL_GUI_COLUMN_TREE=>RELAT_LAST_CHILD
      I_NODE_TEXT      = L_NODE_TEXT
      IS_OUTTAB_LINE   = WA_FINAL1
      IS_NODE_LAYOUT   = LS_NODE
      IT_ITEM_LAYOUT   = LT_ITEM_LAYOUT
    IMPORTING
      E_NEW_NODE_KEY   = P_NODE_KEY.
ENDFORM.                               " add_carrid_line
*&amp;---------------------------------------------------------------------*
*&amp;      Module  USER_COMMAND_9000  INPUT
*&amp;---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE USER_COMMAND_9000 INPUT.
  CASE OK_CODE.
    WHEN 'EXIT' OR 'BACK' OR 'CANC'.
      LEAVE PROGRAM.
    WHEN OTHERS.
      CALL METHOD CL_GUI_CFW=>DISPATCH.
  ENDCASE.
  CLEAR OK_CODE.
  CALL METHOD CL_GUI_CFW=>FLUSH.
ENDMODULE.                 " USER_COMMAND_9000  INPUT
```

SELECTION SCREEN

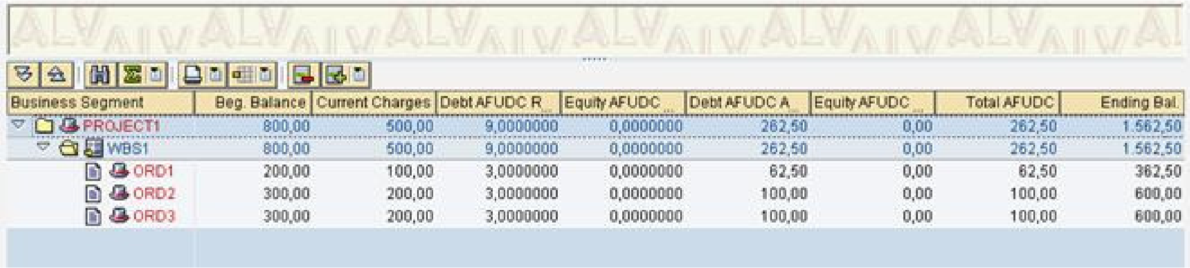


zer14_afudc

SELECT VALUE

S_PROFL		to	
S_PSPNR		to	
P_GJAHR	☑		
V_FM_DT	☑		
V_TO_DT	☑		

OUTPUT



Business Segment	Beg. Balance	Current Charges	Debt AFUDC R	Equity AFUDC	Debt AFUDC A	Equity AFUDC	Total AFUDC	Ending Bal.
PROJECT1	800,00	500,00	9,0000000	0,0000000	262,50	0,00	262,50	1.562,50
WBS1	800,00	500,00	9,0000000	0,0000000	262,50	0,00	262,50	1.562,50
ORD1	200,00	100,00	3,0000000	0,0000000	62,50	0,00	62,50	362,50
ORD2	300,00	200,00	3,0000000	0,0000000	100,00	0,00	100,00	600,00
ORD3	300,00	200,00	3,0000000	0,0000000	100,00	0,00	100,00	600,00

1 Comment

Guest

Ritesh,

Thanks for showing your program. I want to build a simular ALV tree in a full screen mode. Could you descibe/show how your screen 900 looks like?

Kind regards,

Richard meijn