

[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

```

1  *&-----*
2  *& Report  ZER14_AFUDC1
3  *&
4  *&-----*
5  *&
6  *&
7  *&-----*
8  REPORT  ZER14_AFUDC1 MESSAGE-ID ZER14.
9  *&-----*
10 *& DECLARATION OF TYPE POOL.
11 *&
12 *&-----*
13 TYPE-POOLS SLIS.
14 *&-----*
15 *& DECLARATION OF TABLES.
16 *&
17 *&-----*
18 TABLES: AUFK,BKPF,PROJ,TJ02T,PRPS,JCDS,COEP.
19 *&-----*
20 *          TYPE DECLARATION
21 *&-----*
22 TYPES:
23 *FOR aufk TABLE
24     BEGIN OF T_AUFK,
25         AUFNR TYPE AUFK-AUFNR, "ORDER NUMBER
26         OBJNR TYPE AUFK-OBJNR, "OBJECT NUMBER
27         PSPEL TYPE AUFK-PSPEL, "WBS ELEMENT
28         STAT TYPE JCDS-STAT, "OBJECT STATUS
29         UDATE TYPE JCDS-UDATE, "CREATION DATE
30         INACT TYPE JCDS-INACT, "INACTIVE STATUS
31         UTIME TYPE JCDS-UTIME, "TIME CHANGED
32         CHGNR TYPE JCDS-CHGNR, "Change number
33     END OF T_AUFK,
34 *FOR PROJ TABLE
35

```

```

36     BEGIN OF T_PROJ,
37         PSPNR TYPE PROJ-PSPNR, "PROJECT DEFINATION (INITIAL)
38         PSPID TYPE PROJ-PSPID, "PROJECT DEFINATION
39         POST1 TYPE PROJ-POST1, "PROJECT DESCRIPTION
40         PROFL TYPE PROJ-PROFL,
41     END OF T_PROJ,
42 *FOR PRPS TABLE
43     BEGIN OF T_PRPS,
44         PSPNR TYPE PRPS-PSPNR, "WBS ELEMENT
45         OBJNR TYPE PRPS-OBJNR, "OBJECT NUMBER
46         PSPHI TYPE PRPS-PSPHI, "CURRENT NUMBER OF THE APPROPRIATE OBJECT
47         POST1 TYPE PRPS-POST1, "WBS DESCRIPTION
48     END OF T_PRPS,
49 *FOR COEP TABLE
50     BEGIN OF T_COEP,
51         WTGBTR TYPE COEP-WTGBTR, "Total Value in Transaction Currency
52         OBJNR TYPE COEP-OBJNR, "OBJECT NUMBER
53     END OF T_COEP,
54 *FOR COEP TABLE
55     BEGIN OF T_COEP1,
56         WTGBTR TYPE COEP-WTGBTR, "Total Value in Transaction Currency
57         OBJNR TYPE COEP-OBJNR, "OBJECT NUMBER
58     END OF T_COEP1,
59 *FOR TJ02T TABLE
60     BEGIN OF T_TJ02T,
61         ISTAT TYPE TJ02T-ISTAT, "System status
62         TXT04 TYPE TJ02T-TXT04, "ndividual status of an object (short form)
63     END OF T_TJ02T,
64 * FOR T056P TABLE
65     BEGIN OF T_T056P,
66         ZSOLL TYPE T056P-ZSOLL, "Interest Rate
67         DATAB TYPE T056P-DATAB, "Effective-From Date
68     END OF T_T056P,
69 * FOR BKPF TABLE
70     BEGIN OF T_BKPF,
71         GJAHR TYPE BKPF,
72     END OF T_BKPF,
73 *FOR FINAL INTERNAL TABLE
74     BEGIN OF T_FINAL,
75         AUFNR TYPE AUFK-AUFNR, "ORDER NUMBER
76         OBJNR TYPE AUFK-OBJNR, "OBJECT NUMBER
77         PSPEL TYPE AUFK-PSPEL, "WBS ELEMENT
78         POST1 TYPE PROJ-POST1, "PROJECT DESCRIPTION
79         POST11 TYPE PRPS-POST1, "WBS DESCRIPTION
80         STAT TYPE JCDS-STAT, "OBJECT STATUS
81         UDATE TYPE JCDS-UDATE, "CREATION DATE
82         INACT TYPE JCDS-INACT, "INACTIVE STATUS
83         UTIME TYPE JCDS-UTIME, "TIME CHANGED
84         PSPNR TYPE PRPS-PSPNR, "WBS ELEMENT
85         PSPHI TYPE PRPS-PSPHI, "CURRENT NUMBER OF THE APPROPRIATE OBJECT
86         WTGBTR TYPE COEP-WTGBTR, "Total Value in Transaction Currency
87         WTGBTR1 TYPE COEP-WTGBTR, "Total Value in Transaction Currency
88         ZSOLL TYPE T056P-ZSOLL, "Interest Rate

```

```

89         ZSOLL1 TYPE T056P-ZSOLL, "Interest Rate
90         V_VAR1 TYPE P DECIMALS 2, "FOR CALCULATION OF DEBT AMOUNT
91         V_VAR2 TYPE P DECIMALS 2, "FOR CLCULATION OF EQUITY AMOUNT
92         V_TOTAL TYPE P DECIMALS 2, " TOTAL OF V_VAR1 AND V_VAR2
93         V_E_BAL TYPE P DECIMALS 2, "ENDING TOTAL
94         END OF T_FINAL.
95 *&-----*
96 *           PARAMETERS FOR SELECTION-SCREEN
97 *&-----*
98 *           SELECTION SCREEN BLOCK FOR SELECTING VALUE
99 *&-----*
100 SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.
101 SELECT-OPTIONS :
102         S_PROFL FOR PROJ-PROFL,
103         S_PSPNR FOR PROJ-PSPNR.
104 PARAMETERS :
105 *           P_MONAT TYPE BKPF-MONAT OBLIGATORY,
106         P_GJAHR TYPE BKPF-GJAHR OBLIGATORY,
107         V_FM_DT TYPE SY-DATUM OBLIGATORY,
108         V_TO_DT TYPE SY-DATUM OBLIGATORY.
109 SELECTION-SCREEN END OF BLOCK B1.
110 *&-----*
111 *           VARIABLE DECLARATIONS
112 *&-----*
113 DATA: V_ISTAT TYPE TJ02T-ISTAT,
114         V_CAL TYPE P DECIMALS 2, " FOR CALCULATION OF wa_final-zsoll / 12
115         V_CAL1 TYPE P DECIMALS 2. " FOR CALCULATION OF wa_final-zsoll1 / 12
116 *&-----*
117 *           CONTANTS DECLARATION
118 *&-----*
119 CONSTANTS : C_X(1) VALUE 'X', " value used to set X for a field
120         C_REPID(40) VALUE SY-REPID, " report id
121         C_TECO(4) VALUE 'TECO', " object status description
122         C_CHECK(1) VALUE 'X'. " value used to set X for a field
123 CLASS CL_GUI_COLUMN_TREE DEFINITION LOAD.
124 CLASS CL_GUI_CFW DEFINITION LOAD.
125 DATA TREE1 TYPE REF TO CL_GUI_ALV_TREE.
126 DATA MR_TOOLBAR TYPE REF TO CL_GUI_TOOLBAR.
127 INCLUDE <ICON>.
128 INCLUDE BCALV_TOOLBAR_EVENT_RECEIVER.
129 INCLUDE BCALV_TREE_EVENT_RECEIVER.
130 DATA: TOOLBAR_EVENT_RECEIVER TYPE REF TO LCL_TOOLBAR_EVENT_RECEIVER.
131 DATA: IT_FIELD TYPE LVC_T_FCAT, "Fieldcatalog
132         OK_CODE LIKE SY-UCOMM. "OK-Code
133 *-----*
134 *           INTERNAL TABLE
135 *-----*
136 DATA: IT_AUFK TYPE STANDARD TABLE OF T_AUFK, " INTERNAL TABLE OF TYPE I_AUFK
137         IT_PROJ TYPE STANDARD TABLE OF T_PROJ, " INTERNAL TABLE OF TYPE I_PROJ
138         IT_PRPS TYPE STANDARD TABLE OF T_PRPS, " INTERNAL TABLE OF TYPE I_PRPS
139         IT_COEP TYPE STANDARD TABLE OF T_COEP, " INTERNAL TABLE OF TYPE I_COEP
140         IT_COEP1 TYPE STANDARD TABLE OF T_COEP1, " INTERNAL TABLE OF TYPE I_COEP1
141         IT_TJ02T TYPE STANDARD TABLE OF T_TJ02T, " INTERNAL TABLE OF TYPE I_TJ02T

```

```

142         IT_T056P TYPE STANDARD TABLE OF T_T056P, " INTERNAL TABLE OF TYPE I_T056T
143         IT_FINAL TYPE STANDARD TABLE OF T_FINAL, " INTERNAL TABLE OF TYPE I_FIANL
144         IT_BKPF TYPE STANDARD TABLE OF T_BKPF,
145         IT_FINAL1 TYPE STANDARD TABLE OF T_FINAL.
146 *-----
147 *           WORK AREA
148 *-----
149 DATA:WA_AUFK TYPE T_AUFK,
150       WA_PROJ TYPE T_PROJ,
151       WA_PRPS TYPE T_PRPS,
152       WA_COEP TYPE T_COEP,
153       WA_COEP1 TYPE T_COEP1,
154       WA_TJ02T TYPE T_TJ02T,
155       WA_T056P TYPE T_T056P,
156       WA_FINAL TYPE T_FINAL,
157       WA_BKPF TYPE T_BKPF,
158       WA_FINAL1 TYPE T_FINAL.
159 START-OF-SELECTION.
160 *****
161 * perform for passing the hard code value
162 *****
163 * PERFORM HARD_CODE.
164 *****
165 *           PERFORM for get the value
166 *****
167 PERFORM GET_DATA.
168 END-OF-SELECTION.
169 CALL SCREEN 9000.
170 *&-----*
171 *&           Module STATUS_9000 OUTPUT
172 *&-----*
173 *           text
174 *-----*
175 MODULE STATUS_9000 OUTPUT.
176 SET PF-STATUS 'ZMAIN'.
177 * SET TITLEBAR 'xxx'.
178 IF TREE1 IS INITIAL.
179 PERFORM INIT_TREE.
180 ENDIF.
181 CALL METHOD CL_GUI_CFW=>FLUSH.
182 ENDMODULE.           " STATUS_9000 OUTPUT
183 *&-----*
184 *&           Form INIT_TREE
185 *&-----*
186 *           text
187 *-----*
188 * --> p1           text
189 * <-- p2           text
190 *-----*
191 FORM INIT_TREE .
192 IT_FINAL1[] = IT_FINAL[] .
193 * create fieldcatalog for structure sflight
194 PERFORM BUILD_FIELDCATALOG.

```

```

195 * create container for alv-tree
196 DATA: ALV_CONTAINER(30) TYPE C,
197         L_CUSTOM_CONTAINER TYPE REF TO CL_GUI_CUSTOM_CONTAINER.
198 ALV_CONTAINER = 'TREE1'.
199 IF SY-BATCH IS INITIAL.
200     CREATE OBJECT L_CUSTOM_CONTAINER
201     EXPORTING
202         CONTAINER_NAME          = 'ALV_CONTAINER'
203     EXCEPTIONS
204         CNTL_ERROR              = 1
205         CNTL_SYSTEM_ERROR       = 2
206         CREATE_ERROR            = 3
207         LIFETIME_ERROR          = 4
208         LIFETIME_DYNPRO_DYNPRO_LINK = 5.
209 IF SY-SUBRC <> 0.
210     MESSAGE X208(00) WITH 'ERROR'.                "#EC NOTEXT
211 ENDIF.
212 ENDIF.
213 * create tree control
214 CREATE OBJECT TREE1
215     EXPORTING
216         PARENT                  = L_CUSTOM_CONTAINER
217         NODE_SELECTION_MODE     = CL_GUI_COLUMN_TREE=>NODE_SEL_MODE_MULTIPLE
218         ITEM_SELECTION          = SPACE
219         NO_HTML_HEADER          = ''
220         NO_TOOLBAR              = ''
221     EXCEPTIONS
222         CNTL_ERROR              = 1
223         CNTL_SYSTEM_ERROR       = 2
224         CREATE_ERROR            = 3
225         LIFETIME_ERROR          = 4
226         ILLEGAL_NODE_SELECTION_MODE = 5
227         FAILED                  = 6
228         ILLEGAL_COLUMN_NAME     = 7.
229 IF SY-SUBRC <> 0.
230     MESSAGE X208(00) WITH 'ERROR'.                "#EC NOTEXT
231 ENDIF.
232 * create Hierarchy-header
233 DATA L_HIERARCHY_HEADER TYPE TREEV_HHDR.
234 PERFORM BUILD_HIERARCHY_HEADER CHANGING L_HIERARCHY_HEADER.
235 * create info-table for html-header
236 DATA: LT_LIST_COMMENTARY TYPE SLIS_T_LISTHEADER,
237         L_LOGO             TYPE SDYDO_VALUE.
238 * perform build_comment using
239 *         lt_list_commentary
240 *         l_logo.
241 * repid for saving variants
242 DATA: LS_VARIANT TYPE DISVARIANT.
243 LS_VARIANT-REPORT = SY-REPID.
244 BREAK SPCANDXX.
245 REFRESH IT_FINAL.
246 * create emty tree-control
247 CALL METHOD TREE1->SET_TABLE_FOR_FIRST_DISPLAY

```

```

248     EXPORTING
249         IS_HIERARCHY_HEADER = L_HIERARCHY_HEADER
250         IT_LIST_COMMENTARY  = LT_LIST_COMMENTARY
251         I_LOGO              = L_LOGO
252         I_BACKGROUND_ID     = 'ALV_BACKGROUND'
253         I_SAVE              = 'A'
254         IS_VARIANT          = LS_VARIANT
255     CHANGING
256         IT_OUTTAB           = IT_FINAL[] "table must be empty !!
257         IT_FIELDCATALOG    = IT_FIELD[].
258     BREAK SPCANDXX.
259 * create hierarchy
260     PERFORM CREATE_HIERARCHY.
261 * add own functioncodes to the toolbar
262     PERFORM CHANGE_TOOLBAR.
263 * register events
264 * perform register_events.
265 * adjust column_width
266 * call method tree1->COLUMN_OPTIMIZE.
267     ENDFORM.                " INIT_TREE
268 *&-----*
269 *&      Form  build_fieldcatalog
270 *&-----*
271 *
272 *-----*
273 FORM BUILD_FIELDCATALOG .
274     DATA: WA_FIELD TYPE LVC_S_FCAT.
275 * get fieldcatalog
276     CALL FUNCTION 'LVC_FIELDCATALOG_MERGE'
277     EXPORTING
278         I_STRUCTURE_NAME = 'ZSTRUC'
279     CHANGING
280         CT_FIELDCAT      = IT_FIELD.
281     SORT IT_FIELD BY SCRTEXT_L.
282 * change fieldcatalog
283 * data: wa_field type lvc_s_fcat.
284     LOOP AT IT_FIELD INTO WA_FIELD.
285         CASE WA_FIELD-FIELDNAME.
286             WHEN 'POST1' OR 'POST11' OR 'AUFNR'.
287                 WA_FIELD-NO_OUT = 'X'.
288                 WA_FIELD-KEY    = ''.
289             WHEN 'WTGBTR' .
290                 WA_FIELD-OUTPUTLEN = 20.        " output length on screen
291                 WA_FIELD-COLTEXT = TEXT-003. " header information
292                 WA_FIELD-DO_SUM = 'X'.          " DOING SUM
293             WHEN 'WTGBTR1' .
294                 WA_FIELD-OUTPUTLEN = 20.        " output length on screen
295                 WA_FIELD-COLTEXT = TEXT-004. " header information
296                 WA_FIELD-DO_SUM = 'X'.          " DOING SUM
297             WHEN 'ZSOLL' .
298                 WA_FIELD-OUTPUTLEN = 20.        " output length on screen
299                 WA_FIELD-COLTEXT = TEXT-005. " header information
300                 WA_FIELD-DO_SUM = 'X'.          " DOING SUM

```

```

301     WHEN 'ZSOLL1' .
302         WA_FIELD-OUTPUTLEN = 20.          " output length on screen
303         WA_FIELD-COLTEXT = TEXT-006. " header information
304         WA_FIELD-DO_SUM = 'X'.          " DOING SUM
305     WHEN 'V_VAR1' .
306         WA_FIELD-OUTPUTLEN = 20.          " output length on screen
307         WA_FIELD-COLTEXT = TEXT-007. " header information
308         WA_FIELD-DO_SUM = 'X'.          " DOING SUM
309     WHEN 'V_VAR2' .
310         WA_FIELD-OUTPUTLEN = 20.          " output length on screen
311         WA_FIELD-COLTEXT = TEXT-008. " header information
312         WA_FIELD-DO_SUM = 'X'.          " DOING SUM
313     WHEN 'V_TOTAL' .
314         WA_FIELD-OUTPUTLEN = 20.          " output length on screen
315         WA_FIELD-COLTEXT = TEXT-009. " header information
316         WA_FIELD-DO_SUM = 'X'.          " DOING SUM
317     WHEN 'V_E_BAL' .
318         WA_FIELD-OUTPUTLEN = 20.          " output length on screen
319         WA_FIELD-COLTEXT = TEXT-010. " header information
320         WA_FIELD-DO_SUM = 'X'.          " DOING SUM
321     ENDCASE.
322     MODIFY IT_FIELD FROM WA_FIELD.
323     ENDLOOP.
324 ENDFORM..          "build_fieldcatalog
325 *&-----*
326 *&      Form  BUILD_HIERARCHY_HEADER
327 *&-----*
328 *      text
329 *-----*
330 *      <--P_L_HIERARCHY_HEADER  text
331 *-----*
332 FORM BUILD_HIERARCHY_HEADER  CHANGING P_HIERARCHY_HEADER TYPE TREEV_HHDR.
333     P_HIERARCHY_HEADER-HEADING = 'Business Segment'.          "#EC NOTEXT
334     P_HIERARCHY_HEADER-TOOLTIP =
335         'This is the Hierarchy Header !'.  "#EC NOTEXT
336     P_HIERARCHY_HEADER-WIDTH = 30.
337     P_HIERARCHY_HEADER-WIDTH_PIX = ''.
338 ENDFORM.          " BUILD_HIERARCHY_HEADER
339 *&-----*
340 *&      Form  BUILD_COMMENT
341 *&-----*
342 *      text
343 *-----*
344 *      -->P_LT_LIST_COMMENTARY  text
345 *      -->P_L_LOGO  text
346 *-----*
347 FORM BUILD_COMMENT  USING
348     PT_LIST_COMMENTARY TYPE SLIS_T_LISTHEADER
349     P_LOGO              TYPE SDYDO_VALUE.
350     DATA: LS_LINE TYPE SLIS_LISTHEADER.
351 *
352 * LIST HEADING LINE: TYPE H
353 CLEAR LS_LINE.

```

```

354     LS_LINE-TYP  = 'H'.
355 * LS_LINE-KEY:  NOT USED FOR THIS TYPE
356     LS_LINE-INFO = 'ALV-tree-demo: flight-overview'.           "#EC NOTEXT
357     APPEND LS_LINE TO PT_LIST_COMMENTARY.
358 * STATUS LINE: TYPE S
359     CLEAR LS_LINE.
360     LS_LINE-TYP  = 'S'.
361     LS_LINE-KEY  = 'valid until'.                               "#EC NOTEXT
362     LS_LINE-INFO = 'January 29 1999'.                           "#EC NOTEXT
363     APPEND LS_LINE TO PT_LIST_COMMENTARY.
364     LS_LINE-KEY  = 'time'.
365     LS_LINE-INFO = '2.00 pm'.                                   "#EC NOTEXT
366     APPEND LS_LINE TO PT_LIST_COMMENTARY.
367 * ACTION LINE: TYPE A
368     CLEAR LS_LINE.
369     LS_LINE-TYP  = 'A'.
370 * LS_LINE-KEY:  NOT USED FOR THIS TYPE
371     LS_LINE-INFO = 'actual data'.                               "#EC NOTEXT
372     APPEND LS_LINE TO PT_LIST_COMMENTARY.
373     P_LOGO = 'ENJOYSAP_LOGO'.
374     ENDFORM.                " BUILD_COMMENT
375 *&-----*
376 *&      Form  CREATE_HIERARCHY
377 *&-----*
378 *      text
379 *-----*
380 * --> p1      text
381 * <-- p2      text
382 *-----*
383 FORM CREATE_HIERARCHY .
384 * add data to tree
385     DATA: L_PSPNR_KEY TYPE LVC_NKEY,
386            L_PSPNR1_KEY TYPE LVC_NKEY,
387            L_LAST_KEY TYPE LVC_NKEY.
388     SORT IT_FINAL1 BY POST1 POST11 AUFNR.
389     LOOP AT IT_FINAL1 INTO WA_FINAL1.
390         ON CHANGE OF WA_FINAL1-POST1.
391             PERFORM ADD_PSPNR_LINE USING      WA_FINAL1
392                                     ' '
393                                     CHANGING L_PSPNR_KEY.
394         ENDON.
395         ON CHANGE OF WA_FINAL1-POST11.
396             PERFORM ADD_PSPNR1_LINE USING      WA_FINAL1
397                                     L_PSPNR_KEY
398                                     CHANGING L_PSPNR1_KEY.
399         ENDON.
400         PERFORM ADD_COMPLETE_LINE USING      WA_FINAL1
401                                     L_PSPNR1_KEY
402                                     CHANGING L_LAST_KEY.
403         CLEAR WA_FINAL1.
404     ENDLOOP.
405 * calculate totals
406     CALL METHOD TREE1->UPDATE_CALCULATIONS.

```



```

407 * this method must be called to send the data to the frontend
408 CALL METHOD TREE1->FRONTEND_UPDATE.
409 ENDFORM. " CREATE_HIERARCHY
410 *&-----*
411 *& Form GET_DATA
412 *&-----*
413 * text
414 *-----*
415 * --> p1 text
416 * <-- p2 text
417 *-----*
418 FORM GET_DATA .
419 *&-----*
420 * QUERY TO select ISTAT FROM TJ02T INTO IT_TJ02T
421 *&-----*
422 SELECT SINGLE
423 ISTAT
424 FROM TJ02T
425 INTO V_ISTAT
426 WHERE
427 SPRAS = SY-LANGU AND TXT04 = C_TECO.
428 IF SY-SUBRC <> 0.
429 MESSAGE E000.
430 ENDIF.
431 *&-----*
432 * QUERY TO select PSPNR PSPID FROM PROJ INTO IT_PROJ.
433 *&-----*
434 SELECT
435 PSPNR
436 * PSPID
437 POST1
438 FROM PROJ
439 INTO TABLE IT_PROJ
440 WHERE
441 PSPNR IN S_PSPNR
442 AND PROFL IN S_PROFL.
443 IF SY-SUBRC <> 0.
444 MESSAGE E001.
445 ENDIF.
446 *&-----*
447 * QUERY TO select PSPNR OBJNR PSPHI FROM PRPS INTO IT_PRPS.
448 *&-----*
449 SELECT
450 PSPNR
451 OBJNR
452 PSPHI
453 * PSPID
454 POST1
455 FROM PRPS
456 INTO TABLE IT_PRPS
457 FOR ALL ENTRIES IN IT_PROJ
458 WHERE PSPHI = IT_PROJ-PSPNR.
459 IF SY-SUBRC <> 0 .

```

```

460     MESSAGE E002.
461 ENDIF.
462 *&-----*
463 *   QUERY TO select AUFNR OBJNR STAT UDAT INACT UTIME FROM AUFK,JCDS INTO IT_AUFK.
464 *&-----*
465 SELECT
466     A~AUFNR
467     A~OBJNR
468     A~PSPEL
469     B~STAT
470     B~UPDATE
471     B~INACT
472     B~UTIME
473     B~CHGNR
474 FROM
475     AUFK AS A INNER JOIN JCDS AS B
476     ON A~OBJNR = B~OBJNR
477     INTO TABLE IT_AUFK
478     FOR ALL ENTRIES IN IT_PRPS
479     WHERE A~PSPEL = IT_PRPS-PSPNR AND
480           B~STAT = V_ISTAT AND
481           B~UPDATE BETWEEN V_FM_DT AND V_TO_DT
482           AND B~INACT <> C_X.
483 IF SY-SUBRC <> 0 .
484     MESSAGE E003.
485 ENDIF.
486 *&-----*
487 *   QUERY TO select WTGBTR FROM COEP INTO IT_COEP.
488 *&-----*
489 SELECT
490     WTGBTR
491     FROM COEP
492     INTO TABLE IT_COEP
493     FOR ALL ENTRIES IN IT_PRPS
494     WHERE
495         GJAHR = P_GJAHR AND OBJNR = IT_PRPS-OBJNR.
496 IF SY-SUBRC <> 0 .
497     MESSAGE E004.
498 ENDIF.
499 *&-----*
500 *   QUERY TO select WTGBTR FROM COEP INTO IT_COEP.
501 *&-----*
502 SELECT
503     WTGBTR
504     OBJNR
505     FROM COEP
506     INTO TABLE IT_COEP1
507     FOR ALL ENTRIES IN IT_PRPS
508     WHERE
509         GJAHR = P_GJAHR AND OBJNR = IT_PRPS-OBJNR.
510 IF SY-SUBRC <> 0 .
511     MESSAGE E005.
512 ENDIF.

```

```

513 *&-----*
514 *   QUERY TO select ZSOLL FROM T056P INTO IT_T056P.
515 *&-----*
516 SELECT
517     ZSOLL
518     FROM T056P
519     INTO TABLE IT_T056P
520     WHERE REFERENZ = 'ZNWN DEBT' .
521 IF SY-SUBRC <> 0 .
522     MESSAGE E006.
523 ENDIF.
524 *&-----*
525 *   QUERY TO select ZSOLL FROM T056P INTO IT_T056P.
526 *&-----*
527 SELECT
528     ZSOLL
529     FROM T056P
530     INTO TABLE IT_T056P
531     WHERE REFERENZ = 'ZNWNEQUITY' .
532 IF SY-SUBRC <> 0 .
533     MESSAGE E007.
534 ENDIF.
535 BREAK SPCANDXX.
536 LOOP AT IT_PRPS INTO WA_PRPS.
537     WA_FINAL-PSPNR = WA_PRPS-PSPNR.
538     WA_FINAL-OBJNR = WA_PRPS-OBJNR.
539     WA_FINAL-PSPHI = WA_PRPS-PSPHI.
540     WA_FINAL-POST11 = WA_PRPS-POST1.
541     READ TABLE IT_PROJ INTO WA_PROJ WITH KEY PSPNR = WA_PRPS-PSPNR.
542     IF SY-SUBRC = 0.
543         WA_FINAL-POST1 = WA_PROJ-POST1.
544     ELSE.
545         MESSAGE E003.
546     ENDIF.
547     READ TABLE IT_AUFK INTO WA_AUFK WITH KEY PSPEL = WA_PRPS-PSPNR.
548     IF SY-SUBRC = 0.
549         WA_FINAL-AUFNR = WA_AUFK-AUFNR.
550         WA_FINAL-UDATE = WA_AUFK-UDATE.
551         WA_FINAL-INACT = WA_AUFK-INACT.
552         WA_FINAL-UTIME = WA_AUFK-UTIME.
553     ELSE.
554         MESSAGE E003.
555     ENDIF.
556     READ TABLE IT_COEP INTO WA_COEP WITH KEY OBJNR = WA_PRPS-OBJNR.
557     IF SY-SUBRC = 0.
558         WA_FINAL-WTGBTR = WA_COEP-WTGBTR.
559     ELSE.
560         MESSAGE E003.
561     ENDIF.
562     READ TABLE IT_COEP1 INTO WA_COEP1 WITH KEY OBJNR = WA_PRPS-OBJNR.
563     IF SY-SUBRC = 0.
564         WA_FINAL-WTGBTR1 = WA_COEP1-WTGBTR.
565     ELSE.

```

```

566     MESSAGE E003.
567 ENDIF.
568 WA_FINAL-ZSOLL = WA_T056P-ZSOLL.
569 APPEND WA_FINAL TO IT_FINAL. "append wa_final into it_final
570 LOOP AT IT_FINAL INTO WA_FINAL.
571     V_CAL = WA_FINAL-ZSOLL / 12 .
572     V_CAL1 = WA_FINAL-ZSOLL1 / 12 .
573     WA_FINAL-V_VAR1 = ( WA_FINAL-WTGBTR ( V_CAL ) + ( WA_FINAL-WTGBTR1 ) ( V_CAL / 2
574     WA_FINAL-V_VAR2 = ( WA_FINAL-WTGBTR ( V_CAL1 ) + ( WA_FINAL-WTGBTR1 ) ( V_CAL1 ) /
575     WA_FINAL-V_TOTAL = ( WA_FINAL-V_VAR1 + WA_FINAL-V_VAR2 ).
576     WA_FINAL-V_E_BAL = ( WA_FINAL-WTGBTR WA_FINAL-WTGBTR1 WA_FINAL-V_TOTAL ).
577     MODIFY IT_FINAL FROM WA_FINAL INDEX SY-TABIX.
578 ENDLOOP.
579 ENDLOOP.
580 ENDFORM. " GET_DATA
581 *&-----*
582 *& Form CHANGE_TOOLBAR
583 *&-----*
584 * text
585 *-----*
586 * --> p1 text
587 * <-- p2 text
588 *-----*
589 FORM CHANGE_TOOLBAR .
590 * * get toolbar control
591 CALL METHOD TREE1->GET_TOOLBAR_OBJECT
592 IMPORTING
593     ER_TOOLBAR = MR_TOOLBAR.
594 CHECK NOT MR_TOOLBAR IS INITIAL.
595 * add seperator to toolbar
596 CALL METHOD MR_TOOLBAR->ADD_BUTTON
597 EXPORTING
598     FCODE = ''
599     ICON = ''
600     BUTN_TYPE = CNTB_BTTYPE_SEP
601     TEXT = ''
602     QUICKINFO = 'This is a Seperator'. "#EC NOTEXT
603 * add Standard Button to toolbar (for Delete Subtree)
604 CALL METHOD MR_TOOLBAR->ADD_BUTTON
605 EXPORTING
606     FCODE = 'DELETE'
607     ICON = ' (18)'
608     BUTN_TYPE = CNTB_BTTYPE_BUTTON
609     TEXT = ''
610     QUICKINFO = 'Delete subtree'. "#EC NOTEXT
611 * add Dropdown Button to toolbar (for Insert Line)
612 CALL METHOD MR_TOOLBAR->ADD_BUTTON
613 EXPORTING
614     FCODE = 'INSERT_LC'
615     ICON = ' (17)'
616     BUTN_TYPE = CNTB_BTTYPE_DROPDOWN
617     TEXT = ''
618     QUICKINFO = 'Insert Line'. "#EC NOTEXT

```

```

619 * set event-handler for toolbar-control
620 CREATE OBJECT TOOLBAR_EVENT_RECEIVER.
621 SET HANDLER TOOLBAR_EVENT_RECEIVER->ON_FUNCTION_SELECTED
622                                     FOR MR_TOOLBAR.
623 SET HANDLER TOOLBAR_EVENT_RECEIVER->ON_TOOLBAR_DROPDOWN
624                                     FOR MR_TOOLBAR.
625 ENDFORM.                                " CHANGE_TOOLBAR
626 *&-----*
627 *&      Form  ADD_PSPNR_LINE
628 *&-----*
629 *      text
630 *-----*
631 *      -->P_WA_FINAL  text
632 *      -->P_0958  text
633 *      <--P_L_PSPNR_KEY  text
634 *-----*
635 FORM ADD_COMPLETE_LINE USING  WA_FINAL1 TYPE T_FINAL
636                               P_RELAT_KEY TYPE LVC_NKEY
637                               CHANGING P_NODE_KEY TYPE LVC_NKEY.
638 DATA: L_NODE_TEXT TYPE LVC_VALUE. ",
639 *      wa_final2 LIKE it_final.
640 * set item-layout
641 DATA: LT_ITEM_LAYOUT TYPE LVC_T_LAYI,
642       LS_ITEM_LAYOUT TYPE LVC_S_LAYI.
643 LS_ITEM_LAYOUT-T_IMAGE = ' (3P)'.
644 LS_ITEM_LAYOUT-FIELDNAME = TREE1->C_HIERARCHY_COLUMN_NAME.
645 LS_ITEM_LAYOUT-STYLE =
646                     CL_GUI_COLUMN_TREE=>STYLE_INTENSIFD_CRITICAL.
647 APPEND LS_ITEM_LAYOUT TO LT_ITEM_LAYOUT.
648 * add node
649 L_NODE_TEXT = WA_FINAL1-AUFNR.
650 DATA: LS_NODE TYPE LVC_S_LAYN.
651 LS_NODE-N_IMAGE = SPACE.
652 LS_NODE-EXP_IMAGE = SPACE.
653 CALL METHOD TREE1->ADD_NODE
654     EXPORTING
655         I_RELAT_NODE_KEY = P_RELAT_KEY
656         I_RELATIONSHIP   = CL_GUI_COLUMN_TREE=>RELAT_LAST_CHILD
657         I_NODE_TEXT      = L_NODE_TEXT
658         IS_OUTTAB_LINE   = WA_FINAL1
659         IS_NODE_LAYOUT   = LS_NODE
660         IT_ITEM_LAYOUT   = LT_ITEM_LAYOUT
661     IMPORTING
662         E_NEW_NODE_KEY   = P_NODE_KEY.
663 ENDFORM.                                " add_carrid_line
664 *&-----*
665 *&      Form  ADD_PSPNR1_LINE
666 *&-----*
667 *      text
668 *-----*
669 *      -->P_WA_FINAL  text
670 *      -->P_L_PSPNR_KEY  text
671 *      <--P_L_PSPNR1_KEY  text

```

```

672 *-----*
673 FORM ADD_PSPNR1_LINE USING
674             WA_FINAL1 TYPE T_FINAL
675             P_RELAT_KEY TYPE LVC_NKEY
676             CHANGING P_NODE_KEY TYPE LVC_NKEY.
677 DATA: L_NODE_TEXT TYPE LVC_VALUE.  ",
678 *      wa_final2 type t_final.
679 * set item-layout
680 DATA: LT_ITEM_LAYOUT TYPE LVC_T_LAYI,
681       LS_ITEM_LAYOUT TYPE LVC_S_LAYI.
682 LS_ITEM_LAYOUT-T_IMAGE = ' (3Y)'.
683 LS_ITEM_LAYOUT-STYLE   =
684             CL_GUI_COLUMN_TREE=>STYLE_INTENSIFIED.
685 LS_ITEM_LAYOUT-FIELDNAME = TREE1->C_HIERARCHY_COLUMN_NAME.
686 APPEND LS_ITEM_LAYOUT TO LT_ITEM_LAYOUT.
687 * add node
688 L_NODE_TEXT = WA_FINAL1-POST11.
689 DATA: RELAT TYPE INT4.
690 RELAT = CL_GUI_COLUMN_TREE=>RELAT_LAST_CHILD.
691 CALL METHOD TREE1->ADD_NODE
692     EXPORTING
693         I_RELAT_NODE_KEY = P_RELAT_KEY
694         I_RELATIONSHIP   = RELAT
695         I_NODE_TEXT      = L_NODE_TEXT
696         IS_OUTTAB_LINE   = WA_FINAL1
697         IT_ITEM_LAYOUT   = LT_ITEM_LAYOUT
698     IMPORTING
699         E_NEW_NODE_KEY   = P_NODE_KEY.
700 ENDFORM.             " ADD_PSPNR1_LINE
701 *&-----*
702 *&      Form  ADD_PSPNR_LINE
703 *&-----*
704 *      text
705 *-----*
706 *      -->P_WA_FINAL  text
707 *      -->P_0958     text
708 *      <--P_L_PSPNR_KEY text
709 *-----*
710 FORM ADD_PSPNR_LINE USING      WA_FINAL1 TYPE T_FINAL
711                               P_RELAT_KEY TYPE LVC_NKEY
712                               CHANGING P_NODE_KEY TYPE LVC_NKEY.
713 DATA: L_NODE_TEXT TYPE LVC_VALUE.  " ,
714 *      wa_final2 TYPE t_final.
715 * set item-layout
716 DATA: LT_ITEM_LAYOUT TYPE LVC_T_LAYI,
717       LS_ITEM_LAYOUT TYPE LVC_S_LAYI.
718 LS_ITEM_LAYOUT-T_IMAGE = ' (3P)'.
719 LS_ITEM_LAYOUT-FIELDNAME = TREE1->C_HIERARCHY_COLUMN_NAME.
720 LS_ITEM_LAYOUT-STYLE   =
721             CL_GUI_COLUMN_TREE=>STYLE_INTENSIFD_CRITICAL.
722 APPEND LS_ITEM_LAYOUT TO LT_ITEM_LAYOUT.
723 * add node
724 L_NODE_TEXT = WA_FINAL1-POST1.

```

```

725 DATA: LS_NODE TYPE LVC_S_LAYN.
726 LS_NODE-N_IMAGE = SPACE.
727 LS_NODE-EXP_IMAGE = SPACE.
728 CALL METHOD TREE1->ADD_NODE
729 EXPORTING
730     I_RELAT_NODE_KEY = P_RELAT_KEY
731     I_RELATIONSHIP   = CL_GUI_COLUMN_TREE=>RELAT_LAST_CHILD
732     I_NODE_TEXT      = L_NODE_TEXT
733     IS_OUTTAB_LINE   = WA_FINAL1
734     IS_NODE_LAYOUT   = LS_NODE
735     IT_ITEM_LAYOUT   = LT_ITEM_LAYOUT
736 IMPORTING
737     E_NEW_NODE_KEY   = P_NODE_KEY.
738 ENDFORM.                                " add_carrid_line
739 *&-----*
740 *&      Module  USER_COMMAND_9000  INPUT
741 *&-----*
742 *      text
743 *-----*
744 MODULE USER_COMMAND_9000 INPUT.
745     CASE OK_CODE.
746         WHEN 'EXIT' OR 'BACK' OR 'CANC'.
747             LEAVE PROGRAM.
748         WHEN OTHERS.
749             CALL METHOD CL_GUI_CFW=>DISPATCH.
750     ENDCASE.
751     CLEAR OK_CODE.
752     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 similar ALV tree in a full screen mode. Could you descibe/show how your screen 900 looks like?

Kind regards,

Richard meijn

---

### Contact Us

[Privacy](#)

### SAP Help Portal

[Terms of Use](#)

[Legal Disclosure](#)

[Copyright](#)