


```

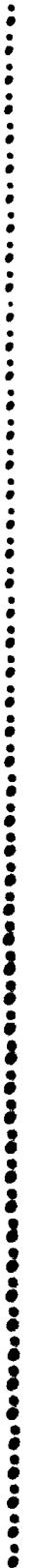
AAAAAA      EEEEEEEEE EEEEEEEEE DDDDDDDD  PPPPPPPP  RRRRRRRR  000000  MM      MM  PPPPPPPP  TTTTTTTTTT
AAAAAA      EEEEEEEEE EEEEEEEEE DDDDDDDD  PPPPPPPP  RRRRRRRR  000000  MM      MM  PPPPPPPP  TTTTTTTTTT
AA          AA      EE      DD      DD  PP      PP  RR      RR  00      00  MMMM  MMMM  PP      PP  TT
AA          AA      EE      DD      DD  PP      PP  RR      RR  00      00  MMMM  MMMM  PP      PP  TT
AA          AA      EE      DD      DD  PP      PP  RR      RR  00      00  MM  MM  MM  PP      PP  TT
AA          AA      EE      DD      DD  PP      PP  RR      RR  00      00  MM  MM  MM  PP      PP  TT
AA          AA      EEEEEEEE DD      DD  PPPPPPPP  RRRRRRRR  00      00  MM      MM  PPPPPPPP  TT
AA          AA      EEEEEEEE DD      DD  PPPPPPPP  RRRRRRRR  00      00  MM      MM  PPPPPPPP  TT
AAAAAAAAAA  EE      DD      DD  PP      RR      RR  00      00  MM      MM  PP      TT
AAAAAAAAAA  EE      DD      DD  PP      RR      RR  00      00  MM      MM  PP      TT
AA          AA      EE      DD      DD  PP      RR      RR  00      00  MM      MM  PP      TT
AA          AA      EE      DD      DD  PP      RR      RR  00      00  MM      MM  PP      TT
AA          AA      EEEEEEEEE DDDDDDDD  PP      RR      RR  000000  MM      MM  PP      TT
AA          AA      EEEEEEEEE DDDDDDDD  PP      RR      RR  000000  MM      MM  PP      TT

```

```

LL          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```



```

1 0001 0 MODULE AED$PROMPT (
2 0002 0     LANGUAGE (BLISS32),
3 0003 0     IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 *   ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 *   TRANSFERRED.
19 0019 1 *
20 0020 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 *   CORPORATION.
23 0023 1 *
24 0024 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1 ++
31 0031 1
32 0032 1 FACILITY:      Miscellaneous utilities
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1     This module contains the routines necessary to support the prompting
37 0037 1     functions of the ACL editor.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1     VAX/VMS operating system, user mode utilities.
42 0042 1
43 0043 1 --
44 0044 1
45 0045 1
46 0046 1 AUTHOR:      L. Mark Pilant      CREATION DATE: 12-Nov-1982  9:50
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1     V03-011 LMP0269      L. Mark Pilant,      28-Jun-1984  15:54
51 0051 1     Remove the AUDIT_JOURNAL from the list of supported ACE
52 0052 1     types.
53 0053 1
54 0054 1     V03-010 LMP0213      L. Mark Pilant,      24-Mar-1984  12:23
55 0055 1     Add support for locking and unlocking the object's ACL.
56 0056 1
57 0057 1     V03-009 LMP0174      L. Mark Pilant,      2-Dec-1983   9:45

```

```

: 58      0058  1  |
: 59      0059  1  |
: 60      0060  1  | V03-008 LMP0172      L. Mark Pilant,      28-Nov-1983  12:11
: 61      0061  1  | Numerous bug fixes, support for VT2xx terminals, and a
: 62      0062  1  | session keystroke logger.
: 63      0063  1  |
: 64      0064  1  | V03-007 LMP0161      L. Mark Pilant,      4-Oct-1983  14:18
: 65      0065  1  | Correctly mark the ACE as open for existing ACEs as well as
: 66      0066  1  | new ACEs.
: 67      0067  1  |
: 68      0068  1  | V03-006 LMP0152      L. Mark Pilant,      12-Sep-1983  15:35
: 69      0069  1  | Make the journal name for ALARM and AUDIT ACEs 'SECURITY'.
: 70      0070  1  |
: 71      0071  1  | V03-005 LMP0138      L. Mark Pilant,      17-Aug-1983  14:43
: 72      0072  1  | Correct various bugs that would cause ACCVIOs during prompt
: 73      0073  1  | mode input.
: 74      0074  1  |
: 75      0075  1  | V03-004 LMP0123      L. Mark Pilant,      22-Jun-1983  11:44
: 76      0076  1  | Change FLAGS item name to OPTIONS. Also, change OWNER
: 77      0077  1  | access definition to CONTROL.
: 78      0078  1  |
: 79      0079  1  | V03-003 LMP0103      L. Mark Pilant,      20-Apr-1983  14:15
: 80      0080  1  | Add support for HIDDEN and PROTECTED ACEs. Also do misc
: 81      0081  1  | fixups.
: 82      0082  1  |
: 83      0083  1  | V03-002 LMP0081      L. Mark Pilant,      22-Feb-1983  13:12
: 84      0084  1  | Don't call SELECTITEM in the size is zero and the PROMPT
: 85      0085  1  | flag is not set.
: 86      0086  1  |
: 87      0087  1  | V03-001 LMP0074      L. Mark Pilant,      21-Jan-1983  12:37
: 88      0088  1  | Fix several bugs that appeared in prompt mode.
: 89      0089  1  |
: 90      0090  1  | **
: 91      0091  1  |
: 92      0092  1  | LIBRARY 'SYSSLIBRARY:LIB.L32';
: 93      0093  1  | LIBRARY 'SYSSLIBRARY:TPAMAC.L32';
: 94      0094  1  | REQUIRE 'SRCS:ACLEDTDEF';

```

SPECIAL NOTE:

96 0547 1
97 0548 1
98 0549 1
99 0550 1
100 0551 1
101 0552 1
102 0553 1
103 0554 1
104 0555 1
105 0556 1
106 0557 1
107 0558 1
108 0559 1
109 0560 1
110 0561 1
111 0562 1
112 0563 1
113 0564 1
114 0565 1
115 0566 1
116 0567 1
117 0568 1
118 0569 1
119 0570 1
120 0571 1
121 0572 1
122 0573 1
123 0574 1
124 0575 1
125 0576 1
126 0577 1
127 0578 1
128 0579 1
129 0580 1
130 0581 1

Since this module depends very much upon the actual format of an ACE, it is absolutely imperative that the routines within this module track that format.

As an aid to determining what portions of the ACE may be defaulted, and what portions may not, the ACE is divided up into items and fields. A field is that portion of the ACE text between commas (or between parens if at the beginning or end of the ACE). An item is that portion of a field between plus signs (or an equal sign on the left and a comma on the right).

Following below are the current ACE types and the field numbers for the various portions of the ACE.

Type: ACESC_KEYID, ACESC_AUDIT, and ACESC_ALARM

(type_keyword, OPT = D + H + P + N , ACC = R + W + E + D + C + S + F + N
^-----0-----^ 1 2 3 4 5 6 7 8 9 1 1 1 1 1
0 1 2 3 4

Type: ACESC_BIJNL, ACESC_AIJNL, and ACESC_ATJNL

(journal_type JOURNAL=name)
^-----0-----^

Type: ACESC_INFO and ACESC_JNLID

Not applicable since these ACE types cannot be entered.

Type: ACESC_DIRDEF

(DEFAULT_PROT, SYSTEM:aaaa, OWNER:bbbb, GROUP:cccc, WORLD:dddd)
^-----0-----^ ^-----1-----^ ^-----2-----^ ^-----3-----^ ^-----4-----^

```
: 132      0582 1 FORWARD ROUTINE
: 133      0583 1      AED_SELECTFIELD : NOVALUE,      ! Select next ACE field
: 134      0584 1      AED_SELECTITEM  : NOVALUE,      ! Select next ACE item
: 135      0585 1      AED_SETACETYPE  : NOVALUE;     ! Set ACE type text
: 136      0586 1
: 137      0587 1 EXTERNAL ROUTINE
: 138      0588 1      AED_POSITION,      ! Position to selected line
: 139      0589 1      AED_SEGSPLIT,     ! Split line segment in two
: 140      0590 1      AED_PUTOUTPUT;    ! General purpose output routine
```

```
142 0591 1 ! ACE item parsing tables.
143 0592 1
144 0593 1 ! Macro to define an entry in the item tables.
145 0594 1
146 0595 1 MACRO
147 0596 1     ENTRY (STRING, VALUE) =
148 0597 1         %CHARCOUNT (STRING),
149 0598 1         UPLIT (STRING),
150 0599 1         VALUE
151 0600 1         %:
152 0601 1
153 0602 1 ! Item tables for parsing and prompting.
154 0603 1
155 0604 1 PSECT PLIT=AED_ITEM_PLIT(OVERLAY,ADDRESSING_MODE(LONG_RELATIVE),ALIGN(0));
156 0605 1
157 0606 1 BIND
158 0607 1
159 0608 1 ! Legal ACE type names. This must parallel the definitions in $ACEDEF,
160 0609 1 ! $PARSE_ACL, and $FORMAT_ACL. The first entry is repeated because the
161 0610 1 ! ACE type codes start at 1, whereas it is possible for 0 to occur during
162 0611 1 ! prompting.
163 0612 1
164 0613 1     ITEM_TYPE      = UPLIT (ENTRY ('IDENTIFIER=', ACESC_KEYID),
165 0614 1         ENTRY ('BI_JOURNAL_NAME=', ACESC_BIJNL),
166 0615 1         ENTRY ('AI_JOURNAL_NAME=', ACESC_AIJNL),
167 0616 1         ENTRY ('AT_JOURNAL_NAME=', ACESC_ATJNL),
168 0617 1         ENTRY ('AUDIT_JOURNAL=SECURITY', ACESC_AUDIT),
169 0618 1         ENTRY ('ALARM_JOURNAL=SECURITY', ACESC_ALARM),
170 0619 1         ENTRY ('INFORMATION', ACESC_INFO),           ! Place holder
171 0620 1         ENTRY ('RMS_JOURNAL_ID', ACESC_JNLID),       ! Place holder
172 0621 1         ENTRY ('DEFAULT_PROTECTION', ACESC_DIRDEF)
173 0622 1         ) : BLOCKVECTOR [,12,BYTE],
174 0623 1
175 0624 1 ! Legal flag names. The wild group and member flags are set based upon
176 0625 1 ! the format of the UIC.
177 0626 1
178 0627 1     ITEM_FLAG      = UPLIT (ENTRY ('DEFAULT+', ACESM_DEFAULT),           ! Assumed to be first
179 0628 1         ENTRY ('HIDDEN+', ACESM_HIDDEN),
180 0629 1         ENTRY ('PROTECTED+', ACESM_PROTECTED),
181 0630 1         ENTRY ('NOPROPAGATE+', ACESM_NOPROPAGATE),
182 0631 1         ENTRY ('NONE+', 0)                                           ! Assumed to be last
183 0632 1         ) : BLOCKVECTOR [,12,BYTE],
184 0633 1
185 0634 1 ! Legal access right names.
186 0635 1
187 0636 1     ITEM_ACCESS    = UPLIT (ENTRY ('READ+', ACESM_READ),
188 0637 1         ENTRY ('WRITE+', ACESM_WRITE),
189 0638 1         ENTRY ('EXECUTE+', ACESM_EXECUTE),
190 0639 1         ENTRY ('DELETE+', ACESM_DELETE),
191 0640 1         ENTRY ('CONTROL+', ACESM_CONTROL),
192 0641 1         ENTRY ('SUCCESS+', ACESM_SUCCESS),
193 0642 1         ENTRY ('FAILURE+', ACESM_FAILURE),
194 0643 1         ENTRY ('NONE+', 0)                                           ! Assume to be last
195 0644 1         ) : BLOCKVECTOR [,12,BYTE];
196 0645 1
197 0646 1 ! Set up constants for the maximum indices for the TYPE, FLAG, and ACCESS
198 0647 1 ! fields.
```

```

: 199      0648 1
: 200      0649 1 LITERAL
: 201      0650 1 MAX_TYPE_INDEX      = 8;      ! Max type field index
: 202      0651 1 MAX_FLAG_INDEX      = 3;      ! Max flag field index
: 203      0652 1 MAX_ACCESS_INDEX     = 7;      ! Max access field index
: 204      0653 1
: 205      0654 1 PSECT PLIT=$PLITS(CONCATENATE,ADDRESSING_MODE(WORD_RELATIVE),ALIGN(2));
```



```

207 0655 1 GLOBAL ROUTINE AED_SELECTFIELD (INDEX) : NOVALUE =
208 0656 1
209 0657 1 !++
210 0658 1
211 0659 1 FUNCTIONAL DESCRIPTION:
212 0660 1
213 0661 1 This routine incrementally selects the various fields for an ACE.
214 0662 1 The field selection is based upon the ACE type. Because this routine
215 0663 1 assumes to know the format of the ACE text, it is necessary to
216 0664 1 insure that it tracks the routine LIB$FORMACL and LIB$PARSACL.
217 0665 1
218 0666 1 CALLING SEQUENCE:
219 0667 1 AED_SELECTFIELD (ARG1)
220 0668 1
221 0669 1 INPUT PARAMETERS:
222 0670 1 ARG1: address of a word containing the current buffer position
223 0671 1
224 0672 1 IMPLICIT INPUTS:
225 0673 1 AED_B_FIELD: current field index (position)
226 0674 1
227 0675 1 OUTPUT PARAMETERS:
228 0676 1 none
229 0677 1
230 0678 1 IMPLICIT OUTPUTS:
231 0679 1 AED_B_FIELD: current field index (position)
232 0680 1
233 0681 1 ROUTINE VALUE:
234 0682 1 none
235 0683 1
236 0684 1 SIDE EFFECTS:
237 0685 1 none
238 0686 1
239 0687 1 !--
240 0688 1
241 0689 2 BEGIN
242 0690 2
243 0691 2 MAP
244 0692 2 INDEX : REF VECTOR [,WORD]; ! Address of the size word
245 0693 2
246 0694 2 LITERAL
247 0695 2 MAX_ACE_CODE = MAXU (ACESC_KEYID,
248 0696 2 ACESC_BIJNL,
249 0697 2 ACESC_AIJNL,
250 0698 2 ACESC_ATJNL,
251 0699 2 ACESC_AUDIT,
252 0700 2 ACESC_ALARM,
253 0701 2 ACESC_INFO,
254 0702 2 ACESC_JNLID,
255 0703 2 ACESC_DIRDEF),
256 0704 2 MIN_ACE_CODE = MINU (ACESC_KEYID,
257 0705 2 ACESC_BIJNL,
258 0706 2 ACESC_AIJNL,
259 0707 2 ACESC_ATJNL,
260 0708 2 ACESC_AUDIT,
261 0709 2 ACESC_ALARM,
262 0710 2 ACESC_INFO,
263 0711 2 ACESC_JNLID,

```

```

: 264      0712      2      ACESC_DIRDEF);
: 265      0713      2
: 266      0714      2 LOCAL
: 267      0715      2      BUFFER_POS,      ! Current buffer position
: 268      0716      2      BUFFER      : REF VECTOR [,BYTE],      ! Address of text storage
: 269      0717      2      ECHO_DESC      : $BBLOCK [DSC$C_S_BLN],      ! Echo str descr
: 270      0718      2      PREV_LINE      : REF $BBLOCK,      ! Address of previous line
: 271      0719      2      TEMP_LINE,      ! Temp display row index
: 272      0720      2      FIELD_KEYWORD      : REF VECTOR [,BYTE];      ! Address of protection field text
: 273      0721      2
: 274      0722      2      ! Note where the actual text is to start.
: 275      0723      2
: 276      0724      2      BUFFER = AED_T_CURLINE[LINE_T_TEXT];
: 277      0725      2
: 278      0726      2      ! Check for the end of the line segment.
: 279      0727      2
: 280      0728      2      IF .AED_T_CURLINE[LINE_W_SIZE] NEQ 0
: 281      0729      2      THEN
: 282      0730      2          BEGIN
: 283      0731      3              IF .BUFFER[.INDEX[0]] EQL ')' OR .BUFFER[.AED_W_ITEMEND + 1] EQL ')'
: 284      0732      3              THEN
: 285      0733      4                  BEGIN
: 286      0734      4                      AED_W_ITEMBEG = AED_W_ITEMEND = INDEX[0] =
: 287      0735      4                          .AED_T_CURLINE[LINE_W_SIZE] - 1;
: 288      0736      4                      RETURN;
: 289      0737      3                      END;
: 290      0738      2              END;
: 291      0739      2
: 292      0740      2      ! Determine the current field index. This is done by counting the number of
: 293      0741      2      ! fields from the beginning of the line. Each field is delimited by either a
: 294      0742      2      ! plus sign or a comma. An equal sign causes the field count to advance to the
: 295      0743      2      ! next major field type (flags or access rights).
: 296      0744      2
: 297      0745      2      BUFFER_POS = 0;
: 298      0746      2      AED_B_FIELD = .AED_T_CURLINE[LINE_B_FIELDST];
: 299      0747      2      UNTIL .BUFFER_POS GEQ .INDEX[0]
: 300      0748      2      DO
: 301      0749      3          BEGIN
: 302      0750      3              IF .BUFFER[.BUFFER_POS] EQL '['
: 303      0751      3              THEN
: 304      0752      4                  BEGIN
: 305      0753      4                      UNTIL .BUFFER[.BUFFER_POS] EQL ']'
: 306      0754      4                      DO BUFFER_POS = .BUFFER_POS + 1;
: 307      0755      3                  END;
: 308      0756      3              IF .BUFFER[.BUFFER_POS] EQL ','
: 309      0757      3              THEN
: 310      0758      4                  BEGIN
: 311      0759      4                      IF .AED_B_FIELD GEQ 1 AND .AED_B_ACETYPE NEQ ACESC_DIRDEF
: 312      0760      4                      THEN AED_B_FIELD = 6
: 313      0761      4                      ELSE AED_B_FIELD = .AED_B_FIELD + 1;
: 314      0762      3                  END;
: 315      0763      3              IF .AED_B_FIELD GEQ 1
: 316      0764      3              THEN
: 317      0765      4                  BEGIN
: 318      0766      4                      IF .BUFFER[.BUFFER_POS] EQL '='
: 319      0767      4                      OR .BUFFER[.BUFFER_POS] EQL '+'
: 320      0768      4                      THEN AED_B_FIELD = .AED_B_FIELD + 1;

```

```
321 0769 3 END;
322 0770 3 BUFFER_POS = .BUFFER_POS + 1;
323 0771 3 END;
324 0772 2
325 0773 2 ! Now for the fun part. Perform anything that depends upon the ACE type.
326 0774 2 ! This is only relevant if something is going to be added to the end of
327 0775 2 ! the current line segment. Otherwise, simply set the boundaries of the
328 0776 2 ! NEXT field.
329 0777 2
330 0778 2 IF .INDEX[0] GEQ .AED_T_CURLINE[LINE_W_SIZE]
331 0779 2 THEN
332 0780 2 BEGIN
333 0781 3
334 0782 3 ! If this is the first field (the ACE type) and the buffer is empty, fill
335 0783 3 ! it with the initial ACE delimiter.
336 0784 3
337 0785 3 IF .INDEX[0] EQL 0 AND .AED_B_FIELD EQL 0
338 0786 3 THEN
339 0787 4 BEGIN
340 0788 4 IF NOT .AED_L_FLAGS[AED_V_PROMPT] THEN RETURN;
341 0789 4 AED_B_ITEM = -1;
342 0790 4 BUFFER[0] = '(';
343 0791 4 INDEX[0] = AED_T_CURLINE[LINE_W_SIZE] = 1;
344 0792 4 END;
345 0793 3
346 0794 3 ! Now build the ACE depending upon the type.
347 0795 3
348 0796 3 CASE .AED_B_ACETYPE FROM 0 TO MAX_ACE_CODE OF
349 0797 3 SET
350 0798 3 [ACESC_KEYID,
351 0799 3 ACESC_AUDIT,
352 0800 3 ACESC_ALARM]:
353 0801 4 BEGIN
354 0802 4
355 0803 4 ! Tie off the current field. This must be done now as the field index may
356 0804 4 ! get bumped.
357 0805 4
358 0806 4 IF .INDEX[0] GTR 0
359 0807 4 THEN
360 0808 5 BEGIN
361 0809 5 IF .BUFFER[.INDEX[0] - 1] NEQ ','
362 0810 5 AND .BUFFER[.INDEX[0] - 1] NEQ '+'
363 0811 5 AND .BUFFER[.INDEX[0] - 1] NEQ '='
364 0812 5 THEN
365 0813 6 BEGIN
366 0814 6 IF .AED_B_FIELD EQL 0 OR .AED_B_FIELD EQL 5
367 0815 6 THEN BUFFER[.INDEX[0]] = ',';
368 0816 6 ELSE IF .AED_B_FIELD EQL 1 OR .AED_B_FIELD EQL 6
369 0817 6 THEN BUFFER[.INDEX[0]] = '=';
370 0818 6 ELSE BUFFER[.INDEX[0]] = '+';
371 0819 6 IF .BUFFER[.INDEX[0]] EQL '=' THEN AED_B_ITEM = -1;
372 0820 6 INDEX[0] = .INDEX[0] + 1;
373 0821 6 AED_T_CURLINE[LINE_W_SIZE] = .AED_T_CURLINE[LINE_W_SIZE] + 1;
374 0822 6 AED_B_FIELD = .AED_B_FIELD + 1;
375 0823 5 END;
376 0824 4 END;
377 0825 4
```

```
378 0826 4 ! If the object is not a directory file, and the current field is the options
379 0827 4 ! field, skip the DEFAULT item. This is a two part operation. The field
380 0828 4 ! is skipped here, and the item is skipped in AED_SELECTITEM.
381 0829 4
382 0830 4     IF NOT .AED_L_FLAGS[AED_V_DIRECTORY]
383 0831 4     AND .AED_B_FIELD GEQ 2 AND .AED_B_FIELD LEQ 5
384 0832 4     THEN AED_B_FIELD = .AED_B_FIELD + 1;
385 0833 4
386 0834 4 ! If at the field for the 'NONE' item, advance to the next field as there have
387 0835 4 ! been items selected for the current field. Also, if items have been
388 0836 4 ! selected for the current field, skip the 'NONE' item, but don't advance the
389 0837 4 ! field position.
390 0838 4
391 0839 4     IF .AED_B_FIELD EQL 5 THEN AED_B_FIELD = 6;
392 0840 4     IF .AED_B_FIELD GEQ 3 AND .AED_B_FIELD LSS 5 AND .AED_B_ITEM EQL MAX_FLAG_INDEX
393 0841 4     THEN AED_B_ITEM = .AED_B_ITEM + 1;
394 0842 4
395 0843 4 ! Skip over SUCCESS and FAILURE items if not ALARM or AUDIT ACEs.
396 0844 4
397 0845 4     IF .AED_B_FIELD GEQ 8 AND .AED_B_ACETYPE EQL ACESC_KEYID AND .AED_B_ITEM EQL 5
398 0846 4     THEN AED_B_ITEM = 7;
399 0847 4
400 0848 4 ! If past the last field, simply return now. This has an implicit NONE item
401 0849 4 ! test for the ACCESS field.
402 0850 4
403 0851 5     IF .AED_B_FIELD GTR (IF .AED_B_ACETYPE EQL ACESC_KEYID THEN 11 ELSE 13)
404 0852 4     THEN RETURN;
405 0853 4
406 0854 4 ! Locate the last segment that has text.
407 0855 4
408 0856 4     PREV_LINE = AED_T_CURLINE;
409 0857 4     TEMP_LINE = .AED_B_LINE;
410 0858 4     UNTIL [.PREV_LINE[LINE_W_SIZE] GTR 0 OR .PREV_LINE[LINE_V_BEGINACE]]
411 0859 4     DO
412 0860 5         BEGIN
413 0861 5         PREV_LINE = .PREV_LINE[LINE_L_BLINK];
414 0862 5         TEMP_LINE = .TEMP_LINE - 1;
415 0863 4         END;
416 0864 4
417 0865 4 ! Determine, based upon the field index, whether or not the keyword must
418 0866 4 ! be added to the line.
419 0867 4
420 0868 4     IF .AED_B_FIELD EQL 1 OR .AED_B_FIELD EQL 6
421 0869 4     THEN
422 0870 5         BEGIN
423 0871 5
424 0872 5 ! Set up the appropriate field text string.
425 0873 5
426 0874 5         CASE .AED_B_FIELD FROM 1 TO 6 OF
427 0875 5         SET
428 0876 5             [1]:     FIELD_KEYWORD = UPLIT (%ASCIC 'OPTIONS=');
429 0877 5             [6]:     FIELD_KEYWORD = UPLIT (%ASCIC 'ACCESS=');
430 0878 5             [INRANGE
431 0879 5             OUTRANGE]: 0;
432 0880 5         TES;
433 0881 5
434 0882 5 ! Make sure previous item tied off properly.
```

```

435 0883 5
436 0884 5
437 0885 5
438 0886 5
439 0887 6
440 0888 6
441 0889 6
442 0890 6
443 0891 6
444 0892 5
445 0893 5
446 0894 5
447 0895 5
448 0896 6
449 0897 6
450 0898 6
451 0899 7
452 0900 7
453 0901 7
454 0902 7
455 0903 6
456 0904 6
457 0905 6
458 0906 6
459 0907 5
460 0908 5
461 0909 5
462 0910 5
463 0911 5
464 0912 5
465 0913 4
466 0914 3
467 0915 3
468 0916 3
469 0917 3
470 0918 3
471 0919 4
472 0920 4
473 0921 4
474 0922 4
475 0923 4
476 0924 3
477 0925 3
478 0926 3
479 0927 4
480 0928 4
481 0929 4
482 0930 4
483 0931 4
484 0932 4
485 0933 4
486 0934 4
487 0935 5
488 0936 5
489 0937 5
490 0938 5
491 0939 4

```

```

IF .VECTOR [PREV_LINE[LINE_T_TEXT],
            .PREV_LINE[LINE_W_SIZE] - 1; .BYTE] EQL '+'
THEN
  BEGIN
    PREV_LINE[LINE_W_SIZE] = .PREV_LINE[LINE_W_SIZE] - 1;
    IF .PREV_LINE EQ[A AED_T_CURLINE
    THEN INDEX[0] = .INDEX[0] - 1
    ELSE AED_W_TOTALSIZE = .AED_W_TOTALSIZE - 1;
    END;
  IF .VECTOR [PREV_LINE[LINE_T_TEXT],
            .PREV_LINE[LINE_W_SIZE] - 1; .BYTE] NEQ ', '
  THEN
    BEGIN
      IF .TEMP_LINE GEQ 1
      THEN
        BEGIN
          AED_POSITION (.PREV_LINE);
          SCR$ERASE_LINE (.AED_B_LINE, .PREV_LINE[LINE_W_SIZE] + 1);
          AED_POSITION (AED_T_CURLINE);
          END;
          BUFFER[INDEX[0]] = ' ';
          INDEX[0] = .INDEX[0] + 1;
          AED_T_CURLINE[LINE_W_SIZE] = .AED_T_CURLINE[LINE_W_SIZE] + 1;
          END;
        CH$MOVE (.FIELD_KEYWORD[0], FIELD_KEYWORD[1], BUFFER[.INDEX[0]]);
        INDEX[0] = .INDEX[0] + .FIELD_KEYWORD[0];
        AED_T_CURLINE[LINE_W_SIZE] = .AED_T_CURLINE[LINE_W_SIZE] + .FIELD_KEYWORD[0];
        AED_B_FIELD = .AED_B_FIELD + 1;
        AED_B_ITEM = -1;
        END;
      END;
    [ACESC_BIJNL,
    ACESS_AIJNL,
    ACESS_ATJNL]:
    BEGIN
      ! If there is anything typed, assume that a journal name is present.
      IF .INDEX[0] GTR 1 THEN RETURN;
      END;
    [ACESC_DIRDEF]:
    BEGIN
      ! Find the previous non-blank line.
      PREV_LINE = AED_T_CURLINE;
      TEMP_LINE = .AED_B_LINE;
      IF NOT .AED_T_CURLINE[LINE_V_BEGINACE]
      THEN DO
        BEGIN
          PREV_LINE = .PREV_LINE[LINE_L_BLINK];
          TEMP_LINE = .TEMP_LINE - 1;
          END
        UNTIL .PREV_LINE[LINE_W_SIZE] GTR 0;

```

```
492 0940 4
493 0941 4 ! If past the last field, simply return now.
494 0942 4
495 0943 4 IF .AED_B_FIELD EQL 4
496 0944 5 AND (IF .AED_T_CURLINE[LINE_W_SIZE] GTR 0
497 0945 5 THEN .VECTOR [AED_T_CURLINE[LINE_T_TEXT], .AED_T_CURLINE[LINE_W_SIZE] - 1; .BYTE] NEQ ', '
498 0946 5 ELSE .VECTOR [PREV_LINE[LINE_T_TEXT], .PREV_LINE[LINE_W_SIZE] - 1; .BYTE] NEQ ', '
499 0947 4 THEN RETURN;
500 0948 4
501 0949 4 ! Make sure previous item tied off properly.
502 0950 4
503 0951 4 IF .VECTOR [PREV_LINE[LINE_T_TEXT],
504 0952 4 .PREV_LINE[LINE_W_SIZE] - 1; .BYTE] NEQ ', '
505 0953 4 THEN
506 0954 5 BEGIN
507 0955 5 IF .TEMP_LINE GEQ 1
508 0956 5 THEN
509 0957 6 BEGIN
510 0958 6 AED_POSITION (.PREV_LINE);
511 0959 6 SCR$ERASE LINE (.AED_B_LINE, .PREV_LINE[LINE_W_SIZE]);
512 0960 6 AED_POSITION (AED_T_CURLINE);
513 0961 6 END;
514 0962 5 BUFFER[INDEX[0]] = ' ';
515 0963 5 INDEX[0] = .INDEX[0] + 1;
516 0964 5 AED_T_CURLINE[LINE_W_SIZE] = .AED_T_CURLINE[LINE_W_SIZE] + 1;
517 0965 5 AED_B_FIELD = .AED_B_FIELD + 1;
518 0966 4 END;
519 0967 4
520 0968 4 ! Set up the appropriate field text string.
521 0969 4
522 0970 4 CASE .AED_B_FIELD FROM 1 TO 4 OF
523 0971 4 SET
524 0972 4 [1]: FIELD_KEYWORD = UPLIT (%ASCIC 'SYSTEM:');
525 0973 4 [2]: FIELD_KEYWORD = UPLIT (%ASCIC 'OWNER:');
526 0974 4 [3]: FIELD_KEYWORD = UPLIT (%ASCIC 'GROUP:');
527 0975 4 [4]: FIELD_KEYWORD = UPLIT (%ASCIC 'WORLD:');
528 0976 4 TES;
529 0977 4
530 0978 4 CH$MOVE (.FIELD_KEYWORD[0], FIELD_KEYWORD[1], BUFFER[INDEX[0]]);
531 0979 4 INDEX[0] = .INDEX[0] + .FIELD_KEYWORD[0];
532 0980 4 AED_T_CURLINE[LINE_W_SIZE] = .AED_T_CURLINE[LINE_W_SIZE] + .FIELD_KEYWORD[0];
533 0981 4 AED_B_FIELD = .AED_B_FIELD + 1;
534 0982 4 END;
535 0983 4 [INRANGE,
536 0984 4 OUTRANGE]: 0;
537 0985 4 TES;
538 0986 4 END;
539 0987 4
540 0988 4 ! Determine the boundaries of the selected field.
541 0989 4
542 0990 4 AED_W_ITEMBEG = .INDEX[0];
543 0991 4 IF .AED_W_ITEMBEG GEQ .AED_T_CURLINE[LINE_W_SIZE]
544 0992 4 THEN
545 0993 5 BEGIN
546 0994 5 AED_W_ITEMEND = .AED_T_CURLINE[LINE_W_SIZE] - 1;
547 0995 5 IF .AED_B_ACETYPE NEQ %C$C_DIRDEF THEN AED_SELECTITEM (.INDEX);
548 0996 5 INDEX[0] = .AED_W_ITEMEND + 1;
```

```

549 0997 3 RETURN;
550 0998 2 END;
551 0999 2
552 1000 2 UNTIL .BUFFER[.AED_W_ITEMBEG] EQL '('
553 1001 2 OR .BUFFER[.AED_W_ITEMBEG] EQL ':'
554 1002 2 OR .BUFFER[.AED_W_ITEMBEG] EQL ')';
555 1003 2 OF (.AED_B_FIELD GTR 0 AND .BUFFER[.AED_W_ITEMBEG] EQL '+')
556 1004 2 DO
557 1005 2 BEGIN
558 1006 2 IF .BUFFER[.AED_W_ITEMBEG] EQL '['
559 1007 2 THEN
560 1008 2 BEGIN
561 1009 2 UNTIL .BUFFER[.AED_W_ITEMBEG] EQL ']'
562 1010 2 DO AED_W_ITEMBEG = .AED_W_ITEMBEG + 1;
563 1011 2 END;
564 1012 2 AED_W_ITEMBEG = .AED_W_ITEMBEG + 1;
565 1013 2 IF .AED_W_ITEMBEG GEQ .AED_T_CURLINE[LINE_W_SIZE]
566 1014 2 THEN
567 1015 2 BEGIN
568 1016 2 AED_W_ITEMEND = .AED_T_CURLINE[LINE_W_SIZE] - 1;
569 1017 2 AED_SELECTITEM (.INDEX);
570 1018 2 INDEX[0] = .AED_W_ITEMEND + 1;
571 1019 2 RETURN;
572 1020 2 END;
573 1021 2 END;
574 1022 2 AED_W_ITEMBEG = .AED_W_ITEMBEG + 1; ! First char past delimiter
575 1023 2
576 1024 2 AED_W_ITEMEND = .AED_W_ITEMBEG;
577 1025 2 IF .AED_W_ITEMEND LSS .AED_T_CURLINE[LINE_W_SIZE]
578 1026 2 THEN
579 1027 2 BEGIN
580 1028 2 UNTIL .BUFFER[.AED_W_ITEMEND] EQL ')'
581 1029 2 OR .BUFFER[.AED_W_ITEMEND] EQL ':'
582 1030 2 OR .BUFFER[.AED_W_ITEMEND] EQL '('
583 1031 2 OR (.AED_B_FIELD GTR 0 AND .BUFFER[.AED_W_ITEMEND] EQL '+')
584 1032 2 DO
585 1033 2 BEGIN
586 1034 2 IF .BUFFER[.AED_W_ITEMEND] EQL '['
587 1035 2 THEN
588 1036 2 BEGIN
589 1037 2 UNTIL .BUFFER[.AED_W_ITEMEND] EQL ']'
590 1038 2 DO AED_W_ITEMEND = .AED_W_ITEMEND + 1;
591 1039 2 END;
592 1040 2 AED_W_ITEMEND = .AED_W_ITEMEND + 1;
593 1041 2 IF .AED_W_ITEMEND GEQ .AED_T_CURLINE[LINE_W_SIZE] THEN EXITLOOP;
594 1042 2 END;
595 1043 2 END;
596 1044 2 IF .AED_W_ITEMEND GTR 0 THEN AED_W_ITEMEND = .AED_W_ITEMEND - 1;
597 1045 2 INDEX[0] = .AED_W_ITEMEND + 1;
598 1046 2
599 1047 2 RETURN;
600 1048 2
601 1049 1 END; ! End of routine AED_SELECTFIELD

```

```
.TITLE AEDSPROMPT
.IDENT \V04-000\

```

```

.PSECT A&D_ITEM_PLIT, NOWRT, NOEXE, OVR, 0
45 4D 41 00 3D 52 45 49 46 49 54 4E 45 44 49 00000 P.AAB: .ASCII \IDENTIFIER=\<0>
4E 5F 4C 41 4E 52 55 4F 4A 5F 49 42 00000C P.AAC: .ASCII \BI_JOURNAL_NAME=\
0001B
45 4D 41 4E 5F 4C 41 4E 52 55 4F 4A 5F 49 41 0001C P.AAD: .ASCII \AI_JOURNAL_NAME=\
3D 0002B
45 4D 41 4E 5F 4C 41 4E 52 55 4F 4A 5F 54 41 0002C P.AAE: .ASCII \AT_JOURNAL_NAME=\
3D 0003B
53 3D 4C 41 4E 52 55 4F 4A 5F 54 49 44 55 41 0003C P.AAF: .ASCII \AUDIT_JOURNAL=SECURITY\<0><0>
00 00 59 54 49 52 55 43 45 0004B
53 3D 4C 41 4E 52 55 4F 4A 5F 4D 52 41 4C 41 00054 P.AAG: .ASCII \ALARM_JOURNAL=SECURITY\<0><0>
00 00 59 54 49 52 55 43 45 00063
00 44 49 00 4E 4F 49 54 41 4D 52 4F 46 4E 49 0006C P.AAH: .ASCII \INFORMATION\<0>
5F 4C 41 4E 52 55 4F 4A 5F 53 4D 52 0007B P.AAI: .ASCII \RMS_JOURNAL_ID\<0><0>
00087
54 43 45 54 4F 52 50 5F 54 4C 55 41 46 45 44 00088 P.AAJ: .ASCII \DEFAULT_PROTECTION\<0><0>
00 00 4E 4F 49 00097
0000000B 0009C P.AAA: .LONG 11
00000000 000A0 .ADDRESS P.AAB
00000010 00000001 000A4 .LONG 1, 16
00000000 000AC .ADDRESS P.AAC
00000010 00000002 000B0 .LONG 2, 16
00000000 000B8 .ADDRESS P.AAD
00000010 00000003 000BC .LONG 3, 16
00000000 000C4 .ADDRESS P.AAE
00000016 00000004 000C8 .LONG 4, 22
00000000 000D0 .ADDRESS P.AAF
00000016 00000005 000D4 .LONG 5, 22
00000000 000DC .ADDRESS P.AAG
0000000B 00000006 000E0 .LONG 6, 11
00000000 000E8 .ADDRESS P.AAH
0000000E 00000007 000EC .LONG 7, 14
00000000 000F4 .ADDRESS P.AAI
00000012 00000008 000F8 .LONG 8, 18
00000000 00100 .ADDRESS P.AAJ
00000009 00104 .LONG 9
00 00 2B 44 2B 54 4C 55 41 46 45 44 00108 P.AAL: .ASCII \DEFAULT+\
2B 45 54 41 47 41 50 4F 52 50 4F 4E 00110 P.AAM: .ASCII \PROTECTED+\<0><0>
00 00 00 2B 44 41 47 41 50 4F 52 50 4F 4E 0011C P.AAN: .ASCII \NOPROPAGATE+\
00 00 00 2B 45 4E 4F 4E 0012B P.AAO: .ASCII \NONE+\<0><0><0>
00000008 00130 P.AAK: .LONG 8
00000000 00134 .ADDRESS P.AAL
0000000A 00000100 00138 .LONG 256, 10
0000000C 00000000 00140 .ADDRESS P.AAM
00000000 00000200 00144 .LONG 512, 12
00000000 0014C .ADDRESS P.AAN
00000005 00000800 00150 .LONG 2048, 5
00000000 00158 .ADDRESS P.AAO
00000000 0015C .LONG 0
00 00 00 2B 44 41 45 52 00160 P.AAQ: .ASCII \READ+\<0><0><0>
00 00 2B 45 54 49 52 57 00168 P.AAR: .ASCII \WRITE+\<0><0>
2B 45 54 55 43 45 58 45 00170 P.AAS: .ASCII \EXECUTE+\
00 2B 45 54 45 4C 45 44 00178 P.AAT: .ASCII \DELETE+\<0>
2B 4C 4F 52 54 4E 4F 43 00180 P.AAU: .ASCII \CONTROL+\
2B 53 53 45 43 43 55 53 00188 P.AAV: .ASCII \SUCCESS+\

```



```

2B 45 52 55 4C 49 41 46 00190 P.AAW: .ASCII: \FAILURE+\
00 00 00 2B 45 4E 4F 4E 00198 P.AAX: .ASCII \NONE+\<0><0><0>
                                001A0 P.AAP: .LONG 5
                                001A4 .ADDRESS P.AAQ
00000006 00000000' 001A8 .LONG 1, 6
                                001B0 .ADDRESS P.AAR
00000008 00000000' 001B4 .LONG 2, 8
                                001BC .ADDRESS P.AAS
00000007 00000000' 001C0 .LONG 4, 7
                                001C8 .ADDRESS P.AAT
00000008 00000000' 001CC .LONG 8, 8
                                001D4 .ADDRESS P.AAU
00000008 00000010' 001D8 .LONG 16, 8
                                001E0 .ADDRESS P.AAV
00000008 00000001' 001E4 .LONG 1, 8
                                001EC .ADDRESS P.AAW
00000005 00000002' 001F0 .LONG 2, 5
                                001F8 .ADDRESS P.AAX
                                001FC .LONG 0

                                .PSECT AED_COMMON,NOEXE, OVR,0

0000 AED_L_FLAGS:
                                .BLKB 4
00004 AED_B_OPTIONS:
                                .BLKB 1
00005 .BLKB 3
00008 AED_L_OBJTYP:
                                .BLKB 4
0000C AED_Q_OBJNAM:
                                .BLKB 8
00014 AED_L_WORSTERR:
                                .BLKB 4
00018 AED_L_PAGEWIDTH:
                                .BLKB 4
0001C AED_L_PAGESIZE:
                                .BLKB 4
00020 AED_B_COLUMN:
                                .BLKB 1
00021 .BLKB 3
00024 AED_B_LINE:
                                .BLKB 1
00025 .BLKB 3
00028 AED_B_SAVE_COL:
                                .BLKB 1
00029 .BLKB 3
0002C AED_B_SAVE_LIN:
                                .BLKB 1
0002D .BLKB 3
00030 AED_Q_LINETABLE:
                                .BLKB 12
0003C AED_L_CURACE:
                                .BLKB 4
00040 AED_L_FIRSTLINE:
                                .BLKB 4
00044 AED_L_LASTLINE:
                                .BLKB 4

```

.....

00048 AED_L_BEGINLINE:
 .BLKB 4
0004C AED_W_INPUTLEN:
 .BLKB 2
0004E .BLKB 2
00050 AED_Q_DEL_ACE:
 .BLKB 8
00058 AED_Q_DEL_LINE:
 .BLKB 8
00060 AED_Q_DEL_WORD:
 .BLKB 8
00068 AED_B_DEL_CHAR:
 .BLKB 1
00069 .BLKB 3
0006C AED_A_ACLBUFFER:
 .BLKB 4
00070 AED_Q_OUTLINE:
 .BLKB 8
00078 AED_W_OBJCHAN:
 .BLKB 2
0007A .BLKB 2
0007C AED_W_TERMIN:
 .BLKB 2
0007E .BLKB 2
00080 AED_W_TERMOUT:
 .BLKB 2
00082 .BLKB 2
00084 AED_W_IOSB:
 .BLKB 8
0008C AED_L_STATUS:
 .BLKB 4
00090 AED_B_FIELD:
 .BLKB 1
00091 .BLKB 3
00094 AED_W_FIELDBEG:
 .BLKB 2
00096 .BLKB 2
00098 AED_W_FIELDEND:
 .BLKB 2
0009A .BLKB 2
0009C AED_B_ITEM:
 .BLKB 1
0009D .BLKB 3
000A0 AED_W_ITEMBEG:
 .BLKB 2
000A2 .BLKB 2
000A4 AED_W_ITEMEND:
 .BLKB 2
000A6 .BLKB 2
000A8 AED_B_ACETYPE:
 .BLKB 1
000A9 .BLKB 3
000AC AED_W_JOURNAL:
 .BLKB 2
000AE .BLKB 2
000B0 AED_T_CURLINE:
 .BLKB 532

```

002C4 AED_W_TOTALSIZE:
          .BLKB 2
002C6          .BLKB 2
002C8 JOURNAL_FAB:
          .BLKB 80
00318 JOURNAL_NAM:
          .BLKB 96
00378 JOURNAL_RAB:
          .BLKB 68
003BC JOURNAL_XABPRO:
          .BLKB 88
00414 JOURNAL_BUFFER:
          .BLKB 10
0041E          .BLKB 2
00420 JOURNAL_INDEX:
          .BLKB 4
00424 RECOVER_FAB:
          .BLKB 80
00474 RECOVER_NAM:
          .BLKB 96
004D4 RECOVER_RAB:
          .BLKB 68
00518 RECOVER_BUFFER:
          .BLKB 10
00522          .BLKB 2
00524 RECOVER_INDEX:
          .BLKB 4

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

```

00 00 00 3D 53 4E 4F 49 54 50 4F 08 00000 P.AAY: .ASCII <8>\OPTIONS=\<0><0><0>
          3D 53 53 45 43 43 41 07 0000C P.AAZ: .ASCII <7>\ACCESS=\
          3A 4D 45 54 53 59 53 07 00014 P.ABA: .ASCII <7>\SYSTEM:\
          00 3A 52 45 4E 57 4F 06 0001C P.ABB: .ASCII <6>\OWNER:\<0>
          00 3A 50 55 4F 52 47 06 00024 P.ABC: .ASCII <6>\GROUP:\<0>
          00 3A 44 4C 52 4F 57 06 0002C P.ABD: .ASCII <6>\WORLD:\<0>

```

```

ITEM_TYPE= P.AAA
ITEM_FLAG= P.AAK
ITEM_ACCESS= P.AAP
.EXTRN CLISGET VALUE, CLISPRESNT
.EXTRN LIB$FREE_VM, LIB$GET_VM
.EXTRN LIB$PARSE, SCR$DOWN_SCROLL
.EXTRN SCR$ERASE LINE, SCR$ERASE PAGE
.EXTRN SCR$SET CURSOR, SCR$SET_SCROLL
.EXTRN SCR$UP_SCROLL, AED$OBJLOCKED
.EXTRN AED$BADKEEP, AED$_LOCATERR
.EXTRN AED$_INIREADERR
.EXTRN AED$_JOUWRITERR
.EXTRN AED$_JOUOPENOUT
.EXTRN AED$_JOUCLOSEOUT
.EXTRN AED$_RECREADERR
.EXTRN AED$_RECOPENIN, AED$ RECLOSEIN
.EXTRN AED$_BADUIC, AED$ BADGRPMEM
.EXTRN AED$_SYNTAX, AED$_BADTYPE
.EXTRN AED$_NOITEMSEL, AED$_MUSTENTER
.EXTRN AED$_INIOPENIN, AED$_INICLOSIN

```

.....

.....

```
.EXTRN AED$_DEFSYNTAX, AED$ NODELETE
.EXTRN AED$_NOMODIFY, AED$_NOHIDDEN
.EXTRN AED$_DUPLICATE, AED$_NOCOMBINE
.EXTRN AED$_NODEFAULT, AED$_NOCTRLCHAR
.EXTRN AED$_NOTFOUND, AED$_CONTROL_C
.EXTRN AED$_ACLUPDATED
.EXTRN AED$_NOCHANGE, AED_POSITION
.EXTRN AED_SEGSPPLIT, AED_PUTOUTPUT
```

```
.PSECT $CODE$,NOWRT,2
```

OFFC 00000

```
.ENTRY AED_SELECTFIELD, Save R2,R3,R4,R5,R6,R7,R8,-; 0655
R9,R10,R11
MOVAB AED_POSITION, R11
MOVAB P.AXY, R10
MOVAB AED_B_FIELD, R9
SUBL2 #8, -SP
MOVAB AED_T_CURLINE+20, BUFFER 0724
MOVZWL AED_T_CURLINE+8, R2 0728
BEQL 2$
MOVL INDEX, R1 0731
MOVZWL (R1), R0
CMPB (R0)[BUFFER], #41
BEQL 1$
MOVZWL AED_W_ITEMEND, R0
CMPB 1(R0)[BUFFER], #41
BNEQ 2$
MOVAB -1(R2), R0 0735
MOVW R0, (R1)
MOVW R0, AED_W_ITEMEND 0734
MOVW R0, AED_W_ITEMBEG
RET 0733
CLRL BUFFER_POS 0745
MOVB AED_T_CURLINE+16, AED_B_FIELD 0746
MOVL INDEX, R7 0747
CMPZV #0, #16, (R7), BUFFER_POS
BLEQ 10$
CMPB (BUFFER_POS)[BUFFER], #91 0750
BNEQ 5$
CMPB (BUFFER_POS)[BUFFER], #93 0753
BEQL 5$
INCL BUFFER_POS 0754
BRB 4$
CMPB (BUFFER_POS)[BUFFER], #44 0756
BNEQ 7$
TSTB AED_B_FIELD 0759
BEQL 6$
CMPB AED_B_ACETYPE, #9
BEQL 6$
MOVB #6, AED_B_FIELD 0760
BRB 7$
INCB AED_B_FIELD 0761
TSTB AED_B_FIELD 0763
BEQL 9$
CMPB (BUFFER_POS)[BUFFER], #61 0766
BEQL 8$
CMPB (BUFFER_POS)[BUFFER], #43 0767
```

```
5B 0000G CF 9E 00002
5A 0000' CF 9E 00007
59 0000' CF 9E 0000C
5E 08 C2 00011
56 34 A9 9E 00014
52 28 A9 3C 00018
28 13 0001C
51 04 AC D0 0001E
50 61 3C 00022
29 6046 91 00025
08 13 00029
50 14 A9 3C 00C?B
29 01 A046 91 0002F
10 12 00034
50 FF A2 9E 00036 1$:
61 50 B0 0003A
14 A9 50 B0 0003D
10 A9 50 B0 00041
04 00045
50 D4 00046 2$:
69 30 A9 90 00048
57 04 AC D0 0004C
10 00 ED 00050 3$:
3F 15 00055
5B 8F 6046 91 00057
08 12 0005C
5D 8F 6046 91 0005E 4$:
04 13 00063
50 D6 00065
F5 11 00067
2C 6046 91 00069 5$:
11 12 0006D
69 95 0006F
08 13 00071
09 18 A9 91 00073
05 13 00077
69 06 90 00079
02 11 0007C
69 96 0007E 6$:
69 95 00080 7$:
0E 13 00082
3D 6046 91 00084
06 13 00088
28 6046 91 0008A
```

50

67

			02	12	0008E		BNEQ	9\$			
			69	96	00090	8\$:	INCB	AED_B_FIELD			0768
			50	D6	00092	9\$:	INCL	BUFFER_POS			0770
			BA	11	00094		BRB	3\$			0747
		52	67	B1	00096	10\$:	CMPW	(R7), R2			0778
			39	1F	00099		BLSSU	14\$			
			67	B5	0009B		TSTW	(R7)			0785
			19	12	0009D		BNEQ	12\$			
			69	95	0009F		TSTB	AED_B_FIELD			
			15	12	000A1		BNEQ	12\$			
			FF71	C9	000A3		TSTB	AED_L_FLAGS+1			0788
			01	19	000A7		BLSS	11\$			
				04	000A9		RET				
	OC	A9	01	8E	000AA	11\$:	MNEGB	#1, AED_B_ITEM			0789
		66	28	90	000AE		MOVB	#40, (BUFFER)			0790
	28	A9	01	B0	000B1		MOVW	#1, AED_T_CURLINE+8			0791
		67	01	B0	000B5		MOVW	#1, (R7)			
		52	18	A9	9A	000B8	12\$:	MOVZBL	AED_B_ACETYPE, R2		0796
		00	52	8F	000BC		CASEB	R2, #0, #9			
0185		09	0234		000C0	13\$:	.WORD	57\$-13\$,-			
0234		0185	0185		000C8			15\$-13\$,-			
		0017	0234		000D0			42\$-13\$,-			
		0017						42\$-13\$,-			
		018B						42\$-13\$,-			
								15\$-13\$,-			
								15\$-13\$,-			
								57\$-13\$,-			
								57\$-13\$,-			
								43\$-13\$			
			021D	31	000D4	14\$:	BRW	57\$			
			67	B5	000D7	15\$:	TSTW	(R7)			0806
			5A	13	000D9		BEQL	22\$			
		50	67	3C	000DB		MOVZWL	(R7), R0			0809
		50	FF	A046	9A	000DE	MOVZBL	-1(R0)[BUFFER], R0			
		2C	50	91	000E3		CMPB	R0, #44			
			4D	13	000E6		BEQL	22\$			
		28	50	91	000E8		CMPB	R0, #43			0810
			48	13	000EB		BEQL	22\$			
		3D	50	91	000ED		CMPB	R0, #61			0811
			43	13	000F0		BEQL	22\$			
		51	69	9A	000F2		MOVZBL	AED_B_FIELD, R1			0814
			05	13	000F5		BEQL	16\$			
		05	51	91	000F7		CMPB	R1, #5			
			0B	12	000FA		BNEQ	17\$			
		50	67	3C	000FC	16\$:	MOVZWL	(R7), R0			0815
		50	56	C0	000FF		ADDL2	BUFFER, R0			
		60	2C	90	00102		MOVB	#44, (R0)			
			1E	11	00105		BRB	20\$			
		01	51	91	00107	17\$:	CMPB	R1, #1			0816
			05	13	0010A		BEQL	18\$			
		06	51	91	0010C		CMPB	R1, #6			
			0B	12	0010F		BNEQ	19\$			
		50	67	3C	00111	18\$:	MOVZWL	(R7), R0			0817
		50	56	C0	00114		ADDL2	BUFFER, R0			
		60	3D	90	00117		MOVB	#61, (R0)			
			09	11	0011A		BRB	20\$			
		50	67	3C	0011C	19\$:	MOVZWL	(R7), R0			0818

		50		56	C0	0011F		ADDL2	BUFFER, R0		
		60		28	90	00122		MOVW	#43, (R0)		
		30		60	91	00125	20\$:	CMPB	(R0), #61		0819
				04	12	00128		BNEQ	21\$		
	OC	A9		01	8E	0012A		MNEGB	#1, AED_B_ITEM		
				67	B6	0012E	21\$:	INCW	(R7)		0820
			28	A9	B6	00130		INCW	AED_T_CURLINE+8		0821
				69	96	00133		INCB	AED_B_FIELD		0822
	OC	FF72	C9	02	E0	00135	22\$:	BBS	#2, AED_L_FLAGS+2, 23\$		0830
			02	69	91	0013B		CMPB	AED_B_FIECD, #2		0831
				07	1F	0013E		BLSSU	23\$		
			05	69	91	00140		CMPB	AED_B_FIELD, #5		
				02	1A	00143		BGTRU	23\$		
				69	96	00145		INCB	AED_B_FIELD		0832
			05	69	91	00147	23\$:	CMPB	AED_B_FIELD, #5		0839
				03	12	0014A		BNEQ	24\$		
			69	06	90	0014C		MOVW	#6, AED_B_FIELD		
			50	69	9A	0014F	24\$:	MOVZBL	AED_B_FIECD, R0		0840
			03	50	91	00152		CMPB	R0, #3		
				0E	1F	00155		BLSSU	25\$		
			05	50	91	00157		CMPB	R0, #5		
				09	1E	0015A		BGEQU	25\$		
			03	OC	A9	91	0015C	CMPB	AED_B_ITEM, #3		
				03	12	00160		BNEQ	25\$		
				OC	A9	96	00162	INCB	AED_B_ITEM		0841
			08	50	91	00165	25\$:	CMPB	R0, #8		0845
				0F	1F	00168		BLSSU	26\$		
			01	52	91	0016A		CMPB	R2, #1		
				0A	12	0016D		BNEQ	26\$		
			05	OC	A9	91	0016F	CMPB	AED_B_ITEM, #5		
				04	12	00173		BNEQ	26\$		
	OC	A9		07	90	00175		MOVW	#7, AED_B_ITEM		0846
			01	52	91	00179	26\$:	CMPB	R2, #1		0851
				05	12	0017C		BNEQ	27\$		
			51	08	D0	0017E		MOVL	#11, R1		
				03	11	00181		BRB	28\$		
			51	0D	D0	00183	27\$:	MOVL	#13, R1		
			51	50	D1	00186	28\$:	CMPB	R0, R1		
				01	15	00189		BLEQ	29\$		
				04	0018B			RET			
			52	20	A9	9E	0018C	29\$:	MOVAB	AED_T_CURLINE, PREV_LINE	0856
			51	94	A9	9A	00190	MOVZBL	AED_B_LINE, TEMP_LINE		0857
				08	A2	B5	00194	30\$:	TSTW	8(PREV_LINE)	0858
				OC	12	00197		BNEQ	31\$		
			08	0A	A2	E8	00199	BLBS	10(PREV_LINE), 31\$		
			52	04	A2	D0	0019D	MOVL	4(PREV_LINE), PREV_LINE		0861
					51	D7	001A1	DECL	TEMP_LINE		0862
					EF	11	001A3	BRB	30\$		0858
			01	50	91	001A5	31\$:	CMPB	R0, #1		0868
				08	13	001A8		BEQL	32\$		
			06	50	91	001AA		CMPB	R0, #6		
				03	13	001AD		BEQL	32\$		
				0142	31	001AF		BRW	57\$		
			01	50	8F	001B2	32\$:	CASEB	R0, #1, #5		0874
0017		05	0017	000E		001B6	33\$:	.WORD	34\$-33\$, -		
	0017		0013	0017		001BE			36\$-33\$, -		
									36\$-33\$, -		

						36\$-33\$,-			
						36\$-33\$,-			
						35\$-33\$			
						36\$			
58			J9	11	001C2	BRB			
			6A	9E	001C4	34\$: MOVAB	P.AAY, FIELD_KEYWORD		0876
			C4	11	001C7	BRB	36\$		
58	0C		AA	9E	001C9	35\$: MOVAB	P.AAZ, FIELD_KEYWORD		0877
50	08		A2	3C	001CD	36\$: MOVZWL	8(PREV_LINE), RO		0885
2B	13	A042	91	001D1		CMPB	19(RO)[PREV_LINE], #43		
			14	12	001D6	BNEQ	38\$		
			A2	B7	001D8	DECW	8(PREV_LINE)		0888
50	20		A9	9E	001DB	MOVAB	AED_T_CURLINE, RO		0889
50			52	D1	001DF	CMPB	PREV_LINE, RO		
			04	12	001E2	BNEQ	37\$		
			67	B7	001E4	DECW	(R7)		0890
			04	11	001E6	BRB	38\$		
		0234	C9	B7	001E8	37\$: DECW	AED_W_TOTALSIZE		0891
50	08		A2	3C	001EC	38\$: MOVZWL	8(PREV_LINE), RO		0894
2C	13	A042	91	001F0		CMPB	19(RO)[PREV_LINE], #44		
			2C	13	001F5	BEQL	40\$		
			51	D5	001F7	TSTL	TEMP_LINE		0897
			1C	15	001F9	BLEQ	39\$		
			52	DD	001FB	PUSHL	PREV_LINE		0900
6B			01	FB	001FD	CALLS	#1, AED_POSITION		
7E	08		A2	3C	00200	MOVZWL	8(PREV_LINE), -(SP)		0901
			6E	D6	00204	INCL	(SP)		
7E	94		A9	9A	00206	MOVZBL	AED_B_LINE, -(SP)		
00000000G	00		02	FB	0020A	CALLS	#2, SCRSEASE_LINE		
			A9	9F	00211	PUSHAB	AED_T_CURLINE		0902
			01	FB	00214	CALLS	#1, AED_POSITION		
6B			67	3C	00217	39\$: MOVZWL	(R7), RO		0904
50			2C	90	0021A	MOVB	#44, (RO)[BUFFER]		
6046			67	B6	0021E	INCW	(R7)		0905
			A9	B6	00220	INCW	AED_T_CURLINE+8		0906
51			68	9A	00223	40\$: MOVZBL	(FIELD_KEYWORD), R1		0908
50			67	3C	00226	MOVZWL	(R7), RO		
6046	01		51	28	00229	MOVCB	R1, 1(FIELD_KEYWORD), (RO)[BUFFER]		
			68	9A	0022F	MOVZBL	(FIELD_KEYWORD), RO		0909
			50	A0	00232	ADDW2	RO, (R7)		
			68	9A	00235	MOVZBL	(FIELD_KEYWORD), RO		0910
50	28		50	A0	00238	ADDW2	RO, AED_T_CURLINE+8		
			69	96	0023C	INCB	AED_B_FIELD		0911
			01	8E	0023E	MNECB	#1, AED_B_ITEM		0912
			00AF	31	00242	41\$: BRW	57\$		0796
			67	B1	00245	42\$: CMPW	(R7), #1		0923
			F8	1B	00248	BLEQU	41\$		
			04	04	0024A	RET			
52	20		A9	9E	0024B	43\$: MOVAB	AED_T_CURLINE, PREV_LINE		0931
51	94		A9	9A	0024F	MOVZBL	AED_B_LINE, TEMP_LINE		0932
0B	2A		A9	E8	00253	BLBS	AED_T_CURLINE+10, 45\$		0933
52	04		A2	D0	00257	44\$: MOVL	4(PREV_LINE), PREV_LINE		0936
			51	D7	0025B	DECL	TEMP_LINE		0937
			08	A2	B5	0025D	TSTW	8(PREV_LINE)	0939
			F5	13	00260	BEQL	44\$		
04			69	91	00262	45\$: CMPB	AED_B_FIELD, #4		0943
			19	12	00265	BNEQ	48\$		
50	28		A9	3C	00267	MOVZWL	AED_T_CURLINE+8, RO		0944

				07	15	0026B		BLEQ	46\$				
		2C		33	A940	91 0026D		CMPB	AED_T_CURLINE+19[RO], #44				0945
						09 11 00272		BRB	47\$				
		50		08	A2	3C 00274	46\$:	MOVZWL	8(PREV_LINE), RO				0946
		2C		13	A042	91 00278		CMPB	19(RO)[PREV_LINE], #44				
						01 13 0027D	47\$:	BEQL	48\$				
						04 0027F		RET					
		50		08	A2	3C 00280	48\$:	MOVZWL	8(PREV_LINE), RO				0952
		2C		13	A042	91 00284		CMPB	19(RO)[PREV_LINE], #44				
						2C 13 00289		BEQL	50\$				
						51 D5 0028B		TSTL	TEMP_LINE				0955
						1A 15 0028D		BLEQ	49\$				
						52 DD 0028F		PUSHL	PREV_LINE				0958
		6B				01 FB 00291		CALLS	#1, AED_POSITION				
		7E		08	A2	3C 00294		MOVZWL	8(PREV_LINE), -(SP)				0959
		7E		94	A9	9A 00298		MOVZBL	AED_B_LINE, -(SP)				
		00000000G	00			02 FB 0029C		CALLS	#2, STRSERASE_LINE				
						20 A9 9F 002A3		PUSHAB	AED_T_CURLINE				0960
						01 FB 002A6		CALLS	#1, AED_POSITION				
		6B				67 3C 002A9	49\$:	MOVZWL	(R7), RO				0962
		50				2C 90 002AC		MOVB	#44, (RO)[BUFFER]				
		6046				67 66 002B0		INCB	(R7)				0963
						28 A9 B6 002B2		INCB	AED_T_CURLINE+8				0964
						69 96 002B5		INCB	AED_B_FIELD				0965
						69 8F 002B7	50\$:	CASEB	AED_B_FIELD, #1, #3				0970
001A	03		01			002BB	51\$:	.WORD	52\$-51\$,-				
	0014		000E		0008				53\$-51\$,-				
									54\$-51\$,-				
									55\$-51\$				
									P.ABA, FIELD_KEYWORD				0972
		58		14	AA	9E 002C3	52\$:	MOVAB					
						10 11 002C7		BRB	56\$				
		58		1C	AA	9E 002C9	53\$:	MOVAB	P.ABB, FIELD_KEYWORD				0973
						0A 11 002CD		BRB	56\$				
		58		24	AA	9E 002CF	54\$:	MOVAB	P.ABC, FIELD_KEYWORD				0974
						04 11 002D3		BRB	56\$				
		58		2C	AA	9E 002D5	55\$:	MOVAB	P.ABD, FIELD_KEYWORD				0975
		51				68 9A 002D9	56\$:	MOVZBL	(FIELD_KEYWORD), R1				0978
		50				67 3C 002DC		MOVZWL	(R7), RO				
		6046	01			51 28 002DF		MOVZBL	(FIELD_KEYWORD), R1				
						68 9A 002E5		MOVZWL	(R7), RO				
						67 3C 002DC		MOVZWL	(R7), RO				
						51 28 002DF		MOVZBL	(FIELD_KEYWORD), R1				
						68 9A 002E5		MOVZWL	(R7), RO				
						50 A0 002E8		ADDW2	RO, (R7)				
						68 9A 002EB		MOVZBL	(FIELD_KEYWORD), RO				0980
						50 A0 002EE		ADDW2	RO, AED_T_CURLINE+8				
						69 96 002F2		INCB	AED_B_FIELD				0981
						67 B0 002F4	57\$:	MOVW	(R7), AED_W_ITEMBEG				0990
						10 A9 B1 002F8		CMPW	AED_W_ITEMBEG, AED_T_CURLINE+8				0991
						0E 1F 002FD		BLSSU	58\$				
						01 A3 002FF		SUBW3	#1, AED_T_CURLINE+8, AED_W_ITEMEND				0994
						18 A9 91 00305		CMPB	AED_B_ACETYPE, #9				0995
						48 12 00309		BNEQ	62\$				
						4D 11 0030B		BRB	63\$				0996
						10 A9 3C 0030D	58\$:	MOVZWL	AED_W_ITEMBEG, RO				1000
						6046 9A 00311		MOVZBL	(RO)[BUFFER], RO				
						50 91 00315		CMPB	RO, #40				
						42 12 00318		BNEQ	64\$				
						50 91 0031A		CMPB	RO, #44				1001
						3D 13 0031D		BEQL	64\$				

			29	50	Y1	0031F	CMPB	RO	#41	1002		
				38	13	00322	BEQL	64\$				
				69	95	00324	TSTB	AED_B_FIELD		1003		
				05	13	00326	BEQL	59\$				
			28	50	91	00328	CMPB	RO	#43			
				2F	13	0032B	BEQL	64\$				
			5B	8F	50	91	0032D	59\$:	CMPB	RO	#91	1006
				10	12	00331	BNEQ	61\$				
				50	3C	00333	60\$:	MOVZWL	AED_W_ITEMBEG, RO	1009		
			5D	8F	6046	91	00337	CMPB	(RO)[BUFFER], #93			
				05	13	0033C	BEQL	61\$				
				10	A9	B6	0033E	INCW	AED_W_ITEMBEG	1010		
				F0	11	00341	BRB	60\$				
				10	A9	B6	00343	61\$:	INCW	AED_W_ITEMBEG	1012	
			28	A9	10	A9	B1	00346	CMPW	AED_W_ITEMBEG, AED_T_CURLINE+8	1013	
				C0	1F	0034B	BLSSU	58\$				
14	A9		28	A9	01	A3	0034D	SUBW3	#1, AED_T_CURLINE+8, AED_W_ITEMEND	1016		
				57	DD	00353	62\$:	PUSHL	R7	1017		
		0000V	CF	01	FB	00355	CALLS	#1, AED_SELECTITEM				
				5A	11	0035A	63\$:	BRB	70\$	1018		
				10	A9	B6	0035C	64\$:	INCW	AED_W_ITEMBEG	1022	
			14	A9	10	A9	B0	0035F	MOVW	AED_W_ITEMBEG, AED_W_ITEMEND	1024	
			28	A9	14	A9	B1	00364	CMPW	AED_W_ITEMEND, AED_T_CURLINE+8	1025	
				43	1E	00369	BGEQU	69\$				
			51	14	A9	3C	0036B	MOVZWL	AED_W_ITEMEND, R1	1028		
			50	6146	9A	0036F	65\$:	MOVZBL	(R1)[BUFFER], RO			
			29	50	91	00373	CMPB	RO	#41			
				36	13	00376	BEQL	69\$				
			2C	50	91	00378	CMPB	RO	#44	1029		
				31	13	0037B	BEQL	69\$				
			28	50	91	0037D	CMPB	RO	#40	1030		
				2C	13	00380	BEQL	69\$				
				69	95	00382	TSTB	AED_B_FIELD		1031		
				05	13	00384	BEQL	66\$				
			28	50	91	00386	CMPB	RO	#43			
				23	13	00389	BEQL	69\$				
			5B	8F	50	91	0038B	66\$:	CMPB	RO	#91	1034
				10	12	0038F	BNEQ	68\$				
				14	A9	3C	00391	67\$:	MOVZWL	AED_W_ITEMEND, RO	1037	
			5D	8F	6046	91	00395	CMPB	(RO)[BUFFER], #93			
				05	13	0039A	BEQL	68\$				
				14	A9	B6	0039C	INCW	AED_W_ITEMEND	1038		
				F0	11	0039F	BRB	67\$				
				14	A9	B6	003A1	68\$:	INCW	AED_W_ITEMEND	1040	
			51	14	A9	3C	003A4	MOVZWL	AED_W_ITEMEND, R1	1041		
			51	28	A9	B1	003A8	CMPW	AED_T_CURLINE+8, R1			
				C1	1A	003AC	BGTRU	65\$				
				14	A9	B5	003AE	69\$:	TSTW	AED_W_ITEMEND	1044	
				03	13	003B1	BEQL	70\$				
				14	A9	B7	003B3	DECW	AED_W_ITEMEND			
67			14	A9	14	A9	B7	003B3	70\$:	ADDW3	#1, AED_W_ITEMEND, (R7)	1045
				01	A1	003B6	RET			1049		
				04	003BB							

: Routine Size: 956 bytes, Routine Base: \$CODE\$ + 0000

```
603 1050 1 GLOBAL ROUTINE AED_SELECTITEM (INDEX) : NOVALUE =
604 1051 1
605 1052 1 |++
606 1053 1
607 1054 1 FUNCTIONAL DESCRIPTION:
608 1055 1
609 1056 1 This routine incrementally selects the various items for an ACE field.
610 1057 1 The item selection is based upon the ACE type. Syntax is checked.
611 1058 1
612 1059 1 CALLING SEQUENCE:
613 1060 1 AED_SELECTITEM (ARG1)
614 1061 1
615 1062 1 INPUT PARAMETERS:
616 1063 1 ARG1: address of a word containing the current buffer position
617 1064 1
618 1065 1 IMPLICIT INPUTS:
619 1066 1 none
620 1067 1
621 1068 1 OUTPUT PARAMETERS:
622 1069 1 none
623 1070 1
624 1071 1 IMPLICIT OUTPUTS:
625 1072 1 AED_B_ITEM: current item index (position)
626 1073 1
627 1074 1 ROUTINE VALUE:
628 1075 1 none
629 1076 1
630 1077 1 SIDE EFFECTS:
631 1078 1 none
632 1079 1
633 1080 1 |--
634 1081 1
635 1082 2 BEGIN
636 1083 2
637 1084 2 MAP
638 1085 2 INDEX : REF VECTOR [,WORD], ! Address of the size word
639 1086 2
640 1087 2 OWN
641 1088 2 JOURNAL_ACES : INITIAL (0); ! 0 = no journaling ACES
642 1089 2 ! 1 = journaling ACES allowed
643 1090 2
644 1091 2 LOCAL
645 1092 2 ITEM_DESC : REF $BLOCK, ! Address of item descriptor
646 1093 2 OLD_ITEM_END ! Where old item name ends
647 1094 2 PARTIAL_LENGTH, ! Size of remaining segment
648 1095 2 BUFFER : REF VECTOR [,BYTE], ! Address of text storage
649 1096 2 MAX_ITEM_VAL; ! Upper limit to ITEM pointer
650 1097 2
651 1098 2 BUFFER = AED_T_CURLINE[LINE_T_TEXT];
652 1099 2
653 1100 2 ! Set the next item index.
654 1101 2
655 1102 2 AED_B_ITEM = .AED_B_ITEM + 1;
656 1103 2
657 1104 2 ! Range check the upper limit based upon the field type. If the upper bound
658 1105 2 ! is exceeded, wrap to zero.
659 1106 2
```

```

: 660 1107 2 IF .AED_B_FIELD EQL 0
: 661 1108 2 THEN MAX_ITEM_VAL = MAX_TYPE_INDEX
: 662 1109 2 ELSE IF .AED_B_FIELD GEQ 2 AND .AED_B_FIELD LEQ 5
: 663 1110 2 THEN MAX_ITEM_VAL = MAX_FLAG_INDEX
: 664 1111 2 ELSE MAX_ITEM_VAL = MAX_ACCESS_INDEX;
: 665 1112 2 IF .AED_B_ITEM GTR .MAX_ITEM_VAL THEN AED_B_ITEM = 0;
: 666 1113 2
: 667 1114 2 ! If this is the first field, set up the ACE type. Also, make sure that
: 668 1115 2 ! INFO and JNLID ACEs are skipped. Lastly, skip the DIRDEF ACE type for
: 669 1116 2 ! non-directory objects.
: 670 1117 2
: 671 1118 2 IF .AED_B_FIELD EQL 0 THEN AED_B_ACETYPE = .ITEM_TYPE[.AED_B_ITEM, ITEM_VALUE];
: 672 1119 2 WHILE .AED_B_ACETYPE EQL ACESC_INFO
: 673 1120 2 OR .AED_B_ACETYPE EQL ACESC_JNLID
: 674 1121 2 OR .AED_B_ACETYPE EQL ACESC_AUDIT
: 675 1122 2 OR (.AED_B_ACETYPE EQL ACESC_DIRDEF AND NOT .AED_L_FLAGS[AED_V_DIRECTORY])
: 676 1123 2 OR (NOT .JOURNAL_ACES AND (.AED_B_ACETYPE EQL ACESC_BIJNL OR
: 677 1124 2 .AED_B_ACETYPE EQL ACESC_AIJNL OR
: 678 1125 2 .AED_B_ACETYPE EQL ACESC_ATJNL))
: 679 1126 2 DO
: 680 1127 2 BEGIN
: 681 1128 2 AED_B_ITEM = .AED_B_ITEM + 1;
: 682 1129 2 IF .AED_B_ITEM GTR .MAX_ITEM_VAL THEN AED_B_ITEM = 0;
: 683 1130 2 AED_B_ACETYPE = .ITEM_TYPE[.AED_B_ITEM, ITEM_VALUE];
: 684 1131 2 END;
: 685 1132 2
: 686 1133 2 ! If beyond the end of the ACE, don't even try for an item.
: 687 1134 2
: 688 1135 2 IF .AED_B_FIELD GTR 6 + .MAX_ITEM_VAL + 1 THEN RETURN;
: 689 1136 2
: 690 1137 2 -----
: 691 1138 2 ! The order of the following three tests is important because of the ordering
: 692 1139 2 ! assumed in the item tables. The first test must be for the SUCCESS and
: 693 1140 2 ! FAILURE items in a KEYID ACE (this test may bump the item index up). The
: 694 1141 2 ! next test must be for the NONE item (this may wrap the item index to 0). The
: 695 1142 2 ! final test must be for the DEFAULT item (this is at item index 0).
: 696 1143 2
: 697 1144 2 ! Skip over the SUCCESS and FAILURE items if not AUDIT or ALARM ACEs.
: 698 1145 2
: 699 1146 2 IF .AED_B_FIELD GEQ 7 AND .AED_B_ACETYPE EQL ACESC_KEYID AND .AED_B_ITEM EQL 5
: 700 1147 2 THEN AED_B_ITEM = 7;
: 701 1148 2
: 702 1149 2 ! If pointing to the NONE item, and there has already been something chosen
: 703 1150 2 ! for the field, skip it.
: 704 1151 2
: 705 1152 2 IF .AED_B_FIELD GEQ 3 AND .AED_B_FIELD LEQ 5 AND .AED_B_ITEM EQL MAX_FLAG_INDEX
: 706 1153 2 OR .AED_B_FIELD GEQ 8 AND (.AED_B_ACETYPE EQL ACESC_KEYID
: 707 1154 2 OR .AED_B_ACETYPE EQL ACESC_AUDIT
: 708 1155 2 OR .AED_B_ACETYPE EQL ACESC_ALARM) AND .AED_B_ITEM EQL MAX_ACCESS_INDEX
: 709 1156 2 THEN AED_B_ITEM = 0;
: 710 1157 2
: 711 1158 2 ! If the object is not a directory, and the item index points to the DEFAULT
: 712 1159 2 ! item, and the field index is within the OPTIONS field, skip to the next
: 713 1160 2 ! item.
: 714 1161 2
: 715 1162 2 IF NOT .AED_L_FLAGS[AED_V_DIRECTORY]
: 716 1163 2 AND .AED_B_FIELD GEQ 2 AND .AED_B_FIELD LEQ 5 AND .AED_B_ITEM EQL 0
```

```

717 1164 2 THEN AED_B_ITEM = .AED_B_ITEM + 1;
718 1165
719 1166 ! This is the end of the carefully ordered tests.
720 1167 !-----
721 1168
722 1169 ! Select the desired item descriptor from the appropriate field name table.
723 1170
724 1171 ITEM_DESC = (SELECTONE .AED_B_FIELD OF
725 1172 SET
726 1173 [0]:          ITEM_TYPE[.AED_B_ITEM, ITEM_DSC_SIZE];
727 1174 [1 TO 5]:    ITEM_FLAG[.AED_B_ITEM, ITEM_DSC_SIZE];
728 1175 [6 TO 14]:  ITEM_ACCESS[.AED_B_ITEM, ITEM_DSC_SIZE];
729 1176 TES);
730 1177
731 1178 ! Replace the old item name with the new one.
732 1179
733 1180 OLD_ITEM_END = MINU (.AED_W_ITEMEND, .AED_T_CURLINE[LINE_W_SIZE] - 1);
734 1181 AED_W_ITEMEND = .AED_W_ITEMBEG + .ITEM_DESC[DSCSW_LENGTH] - 1;
735 1182
736 1183 ! See if it is necessary to wrap the line.
737 1184
738 1185 IF .AED_W_ITEMEND GTR .AED_L_PAGEWIDTH
739 1186 THEN
740 1187 BEGIN
741 1188 AED_SEGSPLIT (%REF (.AED_W_ITEMBEG), 1, 0);
742 1189 AED_W_ITEMEND = .AED_W_ITEMEND - .AED_W_ITEMBEG;
743 1190 OLD_ITEM_END = .OLD_ITEM_END - .AED_W_ITEMBEG;
744 1191 AED_W_ITEMBEG = 0;
745 1192 END;
746 1193
747 1194 ! Replace the item.
748 1195
749 1196 CH$COPY (.AED_T_CURLINE[LINE_W_SIZE] - .OLD_ITEM_END - 1, BUFFER[.OLD_ITEM_END + 1],
750 1197 0
751 1198 511 - .AED_W_ITEMEND, BUFFER[.AED_W_ITEMEND + 1]); ! Remove old name
752 1199 CH$MOVE (.ITEM_DESC[DSCSW_LENGTH], .ITEM_DESC[DSCSA_POINTER],
753 1200 BUFFER[.AED_W_ITEMBEG]); ! Insert new name
754 1201 AED_T_CURLINE[LINE_W_SIZE] = .AED_T_CURLINE[LINE_W_SIZE] - .OLD_ITEM_END + .AED_W_ITEMEND;
755 1202 INDEX[0] = .AED_W_ITEMEND + 1;
756 1203 AED_L_FLAGS[AED_V_MODIFIED] = 1; ! Note ACE has changed
757 1204
758 1205 RETURN 0;
759 1206
760 1207 1 END; ! End of routine AED_SELECTITEM

```

.PSECT \$OWNS,NOEXE,2

00000000 00000 JOURNAL_ACES:
.LONG 0

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

.ENTRY AED_SELECTITEM, Save R2,R3,R4,R5,R6,R7,R8,- ; 1050

05		6A	91	000A4		CMPB	AED_B_ITEM, #5		
		03	12	000A7		BNEQ	11\$		
6A		07	90	000A9		MOVB	#7, AED_B_ITEM		1147
03		52	91	000AC	11\$:	CMPB	R2, #3		1152
		0A	1F	000AF		BLSSU	12\$		
05		52	91	000B1		CMPB	R2, #5		
		05	1A	000B4		BGTRU	12\$		
03		6A	91	000B6		CMPB	AED_B_ITEM, #3		
		1D	13	000B9		BEQL	14\$		
08		52	91	000BB	12\$:	CMPB	R2, #8		1153
		1A	1F	000BE		BLSSU	15\$		
50	0C	AA	9A	000C0		MOVZBL	AED_B_ACETYPE, R0		
01		50	91	000C4		CMPB	R0, #1		
		0A	13	000C7		BEQL	13\$		
05		50	91	000C9		CMPB	R0, #5		1154
		05	13	000CC		BEQL	13\$		
06		50	91	000CE		CMPB	R0, #6		1155
		07	12	000D1		BNEQ	15\$		
07		6A	91	000D3	13\$:	CMPB	AED_B_ITEM, #7		
		02	12	000D6		BNEQ	15\$		
		6A	94	000D8	14\$:	CLRB	AED_B_ITEM		1156
10	FF66	CA	02	E0	15\$:	BBS	#2, AED_L_FLAGS+2, 16\$		1162
		02	52	91	000E0	CMPB	R2, #2		1163
		0B	1F	000E3		BLSSU	16\$		
05		52	91	000E5		CMPB	R2, #5		
		06	1A	000E8		BGTRU	16\$		
		6A	95	000EA		TSTB	AED_B_ITEM		
		02	12	000EC		BNEQ	16\$		
		6A	96	000EE		INCB	AED_B_ITEM		1164
		52	D5	000F0	16\$:	TSTL	R2		1173
		0D	12	000F2		BNEQ	17\$		
50		6A	9A	000F4		MOVZBL	AED_B_ITEM, R0		
50		0C	C4	000F7		MULL2	#12, R0		
58	F8	AB40	9E	000FA		MOVAB	ITEM_TYPE[R0], ITEM_DESC		
		30	11	000FF		BRB	21\$		
		13	15	00101	17\$:	BLEQ	18\$		1174
05		52	91	00103		CMPB	R2, #5		
		0E	1A	00106		BGTRU	18\$		
50		6A	9A	00108		MOVZBL	AED_B_ITEM, R0		
50		0C	C4	0010B		MULL2	#12, R0		
58	008C	CB40	9E	0010E		MOVAB	ITEM_FLAG[R0], ITEM_DESC		
		1B	11	00114		BRB	21\$		
06		52	91	00116	18\$:	CMPB	R2, #6		1175
		05	1F	00119		BLSSU	19\$		
0E		52	91	0011B		CMPB	R2, #14		
		05	1B	0011E		BLEQU	20\$		
58		01	CE	00120	19\$:	MNEGL	#1, ITEM_DESC		
		0C	11	00123		BRB	21\$		
50		6A	9A	00125	20\$:	MOVZBL	AED_B_ITEM, R0		
50		0C	C4	00128		MULL2	#12, R0		
58	00FC	CB40	9E	0012B		MOVAB	ITEM_ACCESS[R0], ITEM_DESC		
51	1C	AA	3C	00131	21\$:	MOVZWL	AED_T_CURLINE+8, R1		1180
		51	D7	00135		DECL	R1		
50	08	AA	3C	00137		MOVZWL	AED_W_ITEMEND, R0		
51		50	D1	0013B		CMPL	R0, RT		
		03	1B	0013E		BLEQU	22\$		
50		51	D0	00140		MOVL	R1, R0		

			52		50	D0	00143	22\$:	MOVL	R0, OLD_ITEM_END		
			50	04	AA	3C	00146		MOVZWL	AED_W_ITEMBEG, R0	1181	
			51		68	3C	0014A		MOVZWL	(ITEM_DESC), R1		
			50		51	C0	0014D		ADDL2	R1, R0		
FF7C	CA	08	AA		01	A3	00150		SUBW3	#1, R0, AED_W_ITEMEND		
		08	AA		00	ED	00155		CMPZV	#0, #16, AED_W_ITEMEND, AED_L_PAGEWIDTH	1185	
			10		1F	15	0015D		BLEQ	23\$		
			7E		01	7D	0015F		MOVQ	#1, -(SP)	1188	
		08	AE		04	AA	3C	00162	MOVZWL	AED_W_ITEMBEG, 8(SP)		
			AE	08	AE	9F	00167		PUSHAB	8(SP)		
		0000G	CF		03	FB	0016A		CALLS	#3, AED_SEGSPLIT		
		08	AA		04	AA	A2	0016F	SUBW2	AED_W_ITEMBEG, AED_W_ITEMEND	1189	
			50		04	AA	3C	00174	MOVZWL	AED_W_ITEMBEG, R0	1190	
			52		50	C2	00178		SUBL2	R0, OLD_ITEM_END		
			59		04	AA	B4	0017B	CLRW	AED_W_ITEMBEG	1191	
			59		1C	AA	3C	0017E	23\$:	MOVZWL	AED_T_CURLINE+8, R9	1196
			51			52	C2	00182	SUBL2	OLD_ITEM_END, R9		
			56		FF	A9	9E	00185	MOVAB	-1(R9), R1		
			8F		08	AA	3C	00189	MOVZWL	AED_W_ITEMEND, R6	1198	
		50	000001FF			56	C3	0018D	SUBL3	R6, #511, R0		
50		00	01 A247			51	2C	00195	MOVCS	R1, 1(OLD_ITEM_END)[BUFFER], #0, R0, 1(R6)-[BUFFER]		
			50		01	A647		0019C				
			50		04	AA	3C	0019F	MOVZWL	AED_W_ITEMBEG, R0	1200	
		6047	04			68	28	001A3	MOVCS	(ITEM_DESC), @4(ITEM_DESC), (R0)[BUFFER]		
		1C	AA			56	A1	001A9	ADDW3	R6, R9, AED_T_CURLINE+8	1201	
		04	BC			01	A1	001AE	ADDW3	#1, R6, @INDEX	1202	
			FF64	CA	80	8F	88	001B3	BISB2	#128, AED_L_FLAGS	1203	
						04	001B9		RET		1207	

; Routine Size: 442 bytes, Routine Base: \$CODE\$ + 03BC

```
762 1208 1 GLOBAL ROUTINE AED_SETACETYPE (INDEX) : NOVALUE =
763 1209 1
764 1210 1 !++
765 1211 1
766 1212 1 FUNCTIONAL DESCRIPTION:
767 1213 1
768 1214 1 This routine selects the text corresponding to the desired ACE type
769 1215 1 based upon the previous ACE type (AED_B_ACETYPE) if available, or the
770 1216 1 current item index (AED_B_ITEM) if not.
771 1217 1
772 1218 1 CALLING SEQUENCE:
773 1219 1 AED_SETACETYPE (ARG1)
774 1220 1
775 1221 1 INPUT PARAMETERS:
776 1222 1 ARG1: address of the word containing the current buffer position
777 1223 1
778 1224 1
779 1225 1 IMPLICIT INPUTS:
780 1226 1 AED_B_ACETYPE: current ACE type if non-zero
781 1227 1 AED_B_ITEM: current item index if above is zero
782 1228 1
783 1229 1 OUTPUT PARAMETERS:
784 1230 1 ARG1: address of the word to contain the new buffer position
785 1231 1
786 1232 1 ROUTINE VALUE:
787 1233 1 none
788 1234 1
789 1235 1 SIDE EFFECTS:
790 1236 1 none
791 1237 1
792 1238 1 --
793 1239 1
794 1240 2 BEGIN
795 1241 2
796 1242 2 MAP
797 1243 2 INDEX : REF VECTOR [,WORD]; ! Address of the size word
798 1244 2
799 1245 2 LOCAL
800 1246 2 BUFFER : REF VECTOR [,BYTE], ! Address of text storage
801 1247 2 ITEM_DESC : REF $BLOCK; ! Address of item descriptor
802 1248 2
803 1249 2 ! Set up the initial delimiter.
804 1250 2
805 1251 2 BUFFER = AED_T_CURLINE[LINE_T_TEXT];
806 1252 2 BUFFER[0] = ' ';
807 1253 2
808 1254 2 ! Determine where the ACE type is to be set from, and set it up.
809 1255 2
810 1256 2 IF .AED_B_ACETYPE EQL 0
811 1257 2 THEN ITEM_DESC = ITEM_TYPE[.AED_B_ACETYPE - 1, ITEM_DSC_SIZE]
812 1258 2 ELSE ITEM_DESC = ITEM_TYPE[.AED_B_ITEM, ITEM_DSC_SIZE];
813 1259 2
814 1260 2 CHSMOVE (.ITEM_DESC[DSC$W_LENGTH], .ITEM_DESC[DSC$A_POINTER], BUFFER[1]);
815 1261 2 AED_T_CURLINE[LINE_W_SIZE] = INDEX[0] = .ITEM_DESC[DSC$W_LENGTH] + 1;
816 1262 2
817 1263 2 RETURN;
818 1264 2
```


: 819 1265 1 END:

: End of routine AED_SETACETYPE

			007C	00000	.ENTRY	AED_SETACETYPE, Save R2,R3,R4,R5,R6	:	1208
	51	0000'	CF	9E 00002	MOVAB	AED_T_CURLINE+20, BUFFER	:	1251
	61		28	90 00007	MOVAB	#40, (BUFFER)	:	1252
	50	0000'	CF	9A 0000A	MOVZBL	AED_B_ACETYPE, R0	:	1256
			0D	12 0000F	BNEQ	1\$:	
	50		0C	C4 00011	MULL2	#12, R0	:	1257
	56	00000000'	EF	40 9E 00014	MOVAB	ITEM_TYPE-12[R0], ITEM_DESC	:	
			10	11 0001C	BRB	2\$:	
	50	0000'	CF	9A 0001E	MOVZBL	AED_B_ITEM, R0	:	1258
	50		0C	C4 00023	MULL2	#12, R0	:	
	56	00000000'	EF	40 9E 00026	MOVAB	ITEM_TYPE[R0], ITEM_DESC	:	
01	A1	04	B6	66 28 0002E	MOVCS	(ITEM_DESC), @4(ITEM_DESC), 1(BUFFER)	:	1260
			50	66 3C 00034	MOVZWL	(ITEM_DESC), R0	:	1261
			50	D6 00037	INCL	R0	:	
	04	BC	50	B0 00039	MOVW	R0, @INDEX	:	
	0000'	CF	50	B0 0003D	MOVW	R0, AED_T_CURLINE+8	:	
			04	00042	RET		:	1265

: Routine Size: 67 bytes, Routine Base: \$CODE\$ + 0576

: 820 1266 1
: 821 1267 1 END
: 822 1268 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
AED_COMMON	1320	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(0)
AED_ITEM_PLIT	512	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, OVR, NOPIC, ALIGN(0)
\$PLITS	52	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	1465	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	4	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	42	0	1000	00:01.8
_\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	0	0	14	00:00.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:AEDSPROMPT/OBJ=OBJ\$:AEDSPROMPT MSRC\$:AEDSPROMPT/UPDATE=(ENHS:AEDSPROMPT)

: Size: 1465 code + 1888 data bytes
: Run Time: 00:33.6
: Elapsed Time: 01:39.8
: Lines/CPU Min: 2266
: Lexemes/CPU-Min: 20342
: Memory Used: 440 pages
: Compilation Complete



0004 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 pattern. Each window shows a different VAX/VMS command or system output. The text is small and difficult to read in many of the windows, but some prominent text includes:

- AEDMESSAG LIS
- AEDPROMPT LIS
- SETACL LIS
- AEDSUBR LIS

The screenshots show various system messages, command prompts, and data listings, typical of a VAX/VMS environment. The overall appearance is that of a dense collection of system logs or command outputs.