

[illegible]

[illegible]


```
1 0001 0 MODULE setaudit ( IDENT = 'V04-000'
2 0002 0 ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL=LONG_RELATIVE)
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
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 FACILITY: SET Command
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module implements the DCL command SET AUDIT.
36 0036 1
37 0037 1 ENVIRONMENT:
38 0038 1
39 0039 1 VAX/VMS operating system, user and kernel mode
40 0040 1
41 0041 1 AUTHOR: Gerry Smith 29-Jun-1983
42 0042 1
43 0043 1 Modified by:
44 0044 1
45 0045 1 V03-007 DAS0001 David Solomon 09-Jul-1984
46 0046 1 Fix truncation errors; make nonexternal refs LONG_RELATIVE.
47 0047 1
48 0048 1 V03-006 RSH0104 R. Scott Hanna 17-Feb-1984
49 0049 1 Fix field test 1 problems, comment out journaling code
50 0050 1 and make changes due to new layout of $NSAEVTDEF.
51 0051 1
52 0052 1 V03-005 RSH0101 R. Scott Hanna 05-Feb-1984
53 0053 1 Temporarily disable SET AUDIT.
54 0054 1
55 0055 1 V03-004 MKL0208 Mary Kay Lyons 30-Nov-1983
56 0056 1 Change ITMLST in CRENVV call to LSTADR.
57 0057 1
```

SETAUDIT
V04-000

K 6
16-Sep-1984 00:41:27 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:08:59 [CLIUTL.SRC]SETAUDIT.B32;1

Page 2
(1)

:	58	0058	1	:
:	59	0059	1	:
:	60	0060	1	:
:	61	0061	1	:
:	62	0062	1	:
:	63	0063	1	:
:	64	0064	1	:
:	65	0065	1	:
:	66	0066	1	:
:	67	0067	1	:
:	68	0068	1	:
:	69	0069	1	--

V03-003 GAS0176 Gerry Smith 30-Aug-1983
Add more comments. Also remove some of the kludges that
were necessary to make journal calls work.

V03-002 GAS0173 Gerry Smith 26-Aug-1983
Remove reference to (now) non-existent literal. sigh...

V03-001 GAS0171 Gerry Smith 24-Aug-1983
Remove reference to mailbox and terminal I/O, add
interactive and remote login/out.


```
.. 71 0070 1 |
.. 72 0071 1 | Include files
.. 73 0072 1 |
.. 74 0073 1 | LIBRARY 'SYSS$LIBRARY:LIB';          ! VAX/VMS common definitions
.. 75 0074 1 |
.. 76 0075 1 |
.. 77 0076 1 | Define some flag bits.  THE FIRST TWO BIT POSITIONS MUST NOT CHANGE.
.. 78 0077 1 |
.. 79 0078 1 | MACRO
.. 80 0079 1 |     v_enable   = 0%,          ! /ENABLE
.. 81 0080 1 |     v_disable  = 1%,          ! /DISABLE
.. 82 0081 1 |     v_alarm    = 2%,          ! /ALARM
.. 83 0082 1 |     v_journal  = 3%,          ! /JOURNAL
.. 84 0083 1 |     v_log      = 4%,          ! /LOG
.. 85 0084 1 |     v_new      = 5%,          ! /NEW_VERSION
.. 86 0085 1 |
.. 87 0086 1 | LITERAL
.. 88 0087 1 |     num_sys_events      = 3,
.. 89 0088 1 |     num_loginout_events = 4,
.. 90 0089 1 |     num_loginout_types  = 8,
.. 91 0090 1 |     num_file_events     = 8,
.. 92 0091 1 |     num_access_types    = 6,
.. 93 0092 1 |
.. 94 0093 1 |     arm$m_all           = arm$m_control OR
.. 95 0094 1 |                           arm$m_delete OR
.. 96 0095 1 |                           arm$m_execute OR
.. 97 0096 1 |                           arm$m_read OR
.. 98 0097 1 |                           arm$m_write,
.. 99 0098 1 |
.. 100 0099 1 |     nsa$m_evt_log_all   = nsa$m_evt_log_bat OR
.. 101 0100 1 |                           nsa$m_evt_log_det OR
.. 102 0101 1 |                           nsa$m_evt_log_dia OR
.. 103 0102 1 |                           nsa$m_evt_log_loc OR
.. 104 0103 1 |                           nsa$m_evt_log_net OR
.. 105 0104 1 |                           nsa$m_evt_log_rem OR
.. 106 0105 1 |                           nsa$m_evt_log_sub;
```



```
: 165      0163 1
: 166      0164 1      access_types      : VECTOR[num_access_types]      ! File access types
: 167      0165 1      INITIAL(%ASCID '.ALL',
: 168      0166 1      %ASCID '.CONTROL',
: 169      0167 1      %ASCID '.DELETE',
: 170      0168 1      %ASCID '.EXECUTE',
: 171      0169 1      %ASCID '.READ',
: 172      0170 1      %ASCID '.WRITE'),
: 173      0171 1
: 174      0172 1      access_bits      : VECTOR[num_access_types]      ! File access bit masks
: 175      0173 1      INITIAL(arm$m_all,
: 176      0174 1      arm$m_control,
: 177      0175 1      arm$m_delete,
: 178      0176 1      arm$m_execute,
: 179      0177 1      arm$m_read,
: 180      0178 1      arm$m_write);
```

```
182 0179 1 |
183 0180 1 | Table of contents
184 0181 1 |
185 0182 1 |
186 0183 1 | FORWARD ROUTINE
187 0184 1 |   set$audit : NOVALUE,
188 0185 1 |   get_values : NOVALUE,
189 0186 1 |   set_bits  : NOVALUE;
190 0187 1 |   start_journal,
191 0188 1 |   new_version,
192 0189 1 |   create_journal;
193 0190 1 |
194 0191 1 |
195 0192 1 | External routines
196 0193 1 |
197 0194 1 | EXTERNAL ROUTINE
198 0195 1 |   str$concat,
199 0196 1 |   cli$get_value,
200 0197 1 |   cli$present;
201 0198 1 |
202 0199 1 |
203 0200 1 | Declare literals defined elsewhere
204 0201 1 |
205 0202 1 | EXTERNAL LITERAL
206 0203 1 |   set$_jnlaccerr,
207 0204 1 |   set$_jnlerr,
208 0205 1 |   set$_newjnl,
209 0206 1 |   set$_jnl$sysdev,
210 0207 1 |   set$_vrsnerr,
211 0208 1 |   set$_newvrsn;
212 0209 1 |
213 0210 1 |
214 0211 1 | Declare the shared messages
215 0212 1 |
216 0213 1 | $SHR_MSGDEF (SET, 119, LOCAL,
217 0214 1 |             (syntax, error));
218 0215 1 |
219 0216 1 |
220 0217 1 | Declare some cells in the exec
221 0218 1 |
222 0219 1 | EXTERNAL
223 0220 1 |   ctl$gq_procpriv : $BBLOCK,
224 0221 1 |   nsa$gr_journvec,
225 0222 1 |   nsa$gr_alarmvec;
226 0223 1 |
```

```
| Main module of SET AUDIT
| Parse ENABLE/DISABLE list
| Set alarm/journal bits
| Start a journal
| Make a new version of the journal file
| Create a new journal
```

```
| Make a string
| Get value from CLI
| See if qualifier is present
```

```
| Error accessing journal
| Error creating journal
| Journal successfully created
| Using SYS$SYSDEVICE for journal
| Error creating new version
| New version created
```



```
228 0224 1 GLOBAL ROUTINE set$audit : NOVALUE =
229 0225 2 BEGIN
230 0226 2
231 0227 2 ++
232 0228 2 Functional description
233 0229 2
234 0230 2 This is the routine for the SET AUDIT command. It is called
235 0231 2 from the SET command processor, and enables/disables security
236 0232 2 alarms and security journals, for various event classes. In
237 0233 2 addition, a new journal file may be created.
238 0234 2
239 0235 2 Inputs
240 0236 2 None
241 0237 2
242 0238 2 Outputs
243 0239 2 None
244 0240 2
245 0241 2 ----
246 0242 2
247 0243 2 LOCAL
248 0244 2 status,
249 0245 2 chan : WORD,
250 0246 2 arglist : VECTOR[3],
251 0247 2 mask : VECTOR[2*nsa$sk_evt_length,BYTE] ! Enable/disable flags
252 0248 2 VOLATILE,
253 0249 2 disk : $BBLOCK[dsc$c_s_bln], ! Journal device name
254 0250 2 flags : BITVECTOR[8] ! Flags to tell what we're doing
255 0251 2 INITIAL(BYTE(0));
256 0252 2
257 0253 2
258 0254 2 See if the user has the SECURITY privilege.
259 0255 2
260 0256 2 IF NOT .ctl$gq_procpriv[prv$v_security]
261 0257 2 THEN
262 0258 2 BEGIN
263 0259 2 SIGNAL(ss$_nosecurity);
264 0260 2 RETURN;
265 0261 2 END;
266 0262 2
267 0263 2
268 0264 2 See if logging is required. Also check for /ALARM or /JOURNAL
269 0265 2
270 0266 2 !IF cli$present(%ASCII 'LOG')
271 0267 2 !THEN flags[v_log] = 1;
272 0268 2
273 0269 2 !IF cli$present(%ASCII 'ALARM')
274 0270 2 !THEN flags[v_alarm] = 1;
275 0271 2
276 0272 2 !IF cli$present(%ASCII 'JOURNAL')
277 0273 2 !THEN flags[v_journal] = 1;
278 0274 2
279 0275 2 !IF cli$present(%ASCII 'NEW_VERSION')
280 0276 2 !THEN flags[v_new] = 1;
281 0277 2
282 0278 2 !$init_dyndesc(disk);
283 0279 2 !cli$get_value(%ASCII 'DEVICE_NAME', disk);
284 0280 2
```

```
285 0281 2
286 0282 2
287 0283 2 See if anything is to be enabled or disabled.
288 0284 2
289 0285 2 CH$FILL(0, %ALLOCATION(mask), mask); ! Zero the masks
290 0286 2
291 0287 2 INCR i FROM 0 TO 1 DO ! Loop thru, once for enable,
292 0288 2 BEGIN ! once for disable.
293 0289 2 IF cli$present(.option_string[i]) ! If something to do,
294 0290 2 THEN
295 0291 2 BEGIN
296 0292 2 flags[i] = 1; ! set a bit saying so, and
297 0293 2 get_values(.option_string[i], ! call routine to put new values
298 0294 2 mask[i*nsa$sk_evt_length]) ! in the appropriate mask.
299 0295 2 END;
300 0296 2 END;
301 0297 2
302 0298 2 If something is supposed to be enabled/disabled, go do it.
303 0299 2
304 0300 2
305 0301 2 IF .flags[v_enable] ! See why ENABLE and
306 0302 2 OR .flags[v_disable] ! DISABLE are first in the
307 0303 2 THEN ! bit list?
308 0304 2 BEGIN ! If something is supposed
309 0305 2 arglist[0] = 2; ! to be set/cleared, set
310 0306 2 arglist[1] = mask; ! up the argument list
311 0307 2 arglist[2] = flags; ! and call the routine
312 0308 2 $CMKRN(LROUTIN = set_bits, ARGST = arglist); ! to do the bit tweaking.
313 0309 2 END;
314 0310 2
315 0311 2 If /JOURNAL, then start/create the journal.
316 0312 2
317 0313 2
318 0314 2 IF NOT .flags[v_journal] ! If not a journal,
319 0315 2 THEN RETURN; ! leave.
320 0316 2 IF NOT (.flags[v_enable] ! If not either
321 0317 2 OR .flags[v_new]) ! enabling or new_version,
322 0318 2 THEN RETURN; ! leave.
323 0319 2 arglist[0] = 1; ! Otherwise set up the
324 0320 2 arglist[1] = chan; ! argument list and
325 0321 2 status = $CMEXEC(ROUTIN = start_journal, ! call the routine to
326 0322 2 ARGST = arglist); ! start the journal.
327 0323 2
328 0324 2
329 0325 2 If there was an error trying to start the journal, tell the
330 0326 2 user and attempt to create a new one.
331 0327 2
332 0328 2 IF NOT .status
333 0329 2 THEN
334 0330 2 BEGIN
335 0331 2 SIGNAL(set$jnlaccerr, 0, ! Error trying to start journal
336 0332 2 .status); ! and here's why
337 0333 2 SIGNAL(set$jnl(sysdev); ! and that we'll try on SYSSYSDEVICE
338 0334 2 arglist[0] = 2;
339 0335 2 arglist[1] = chan;
340 0336 2 arglist[2] = disk;
341 0337 2 status = $CMEXEC(ROUTIN = create_journal,
```



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

.PSECT      SPLITS,NOWRT,NOEXE,2

.ASCII      \ENABLE\<0><0>
.LONG        17694726
.ADDRESS     P.AAB
.ASCII      \DISABLE\<0>
.LONG        17694727
.ADDRESS     P.AAD
.ASCII      \ACL\
.LONG        17694724
.ADDRESS     P.AAF
.ASCII      \.AUTHORIZATION\<0><0>

.LONG        17694734
.ADDRESS     P.AAH
.ASCII      \.MOUNT\<0><0>
.LONG        17694726
.ADDRESS     P.AAJ
```

```
4E 49 4B 41 45 52 42 2E 00054 P.AAL: .ASCII \.BREAKIN\
010E0008 0005C P.AAK: .LONG 17694728
00000000 00060 .ADDRESS P.AAL
00 00 4E 49 47 4F 4C 2E 00064 P.AAN: .ASCII \.LOGIN\<0><0>
010E0006 0006C P.AAM: .LONG 17694726
00000000 00070 .ADDRESS P.AAN
00 45 52 55 4C 49 41 46 47 4F 4C 2E 00074 P.AAP: .ASCII \.LOGFAILURE\<0>
010E000B 00080 P.AAO: .LONG 17694731
00000000 00084 .ADDRESS P.AAP
00 54 55 4F 47 4F 4C 2E 00088 P.AAR: .ASCII \.LOGOUT\<0>
010E0007 00090 P.AAQ: .LONG 17694727
00000000 00094 .ADDRESS P.AAR
4C 4C 41 2E 00098 P.AAT: .ASCII \.ALL\
010E0004 0009C P.AAS: .LONG 17694724
00000000 000A0 .ADDRESS P.AAT
00 00 00 44 45 48 43 41 54 45 44 2E 000A4 P.AAV: .ASCII \.DETACHED\<0><0><0>
010E0009 000B0 P.AAU: .LONG 17694729
00000000 000B4 .ADDRESS P.AAV
00 50 55 4C 41 49 44 2E 000B8 P.AAX: .ASCII \.DIALUP\<0>
010E0007 000C0 P.AAW: .LONG 17694727
00000000 000C4 .ADDRESS P.AAX
00 00 4C 41 43 4F 4C 2E 000C8 P.AAZ: .ASCII \.LOCAL\<0><0>
010E0006 000D0 P.AAY: .LONG 17694726
00000000 000D4 .ADDRESS P.AAZ
4B 52 4F 57 54 45 4E 2E 000D8 P.ABB: .ASCII \.NETWORK\
010E0008 000E0 P.ABA: .LONG 17694728
00000000 000E4 .ADDRESS P.ABB
00 45 54 4F 4D 45 52 2E 000E8 P.ABD: .ASCII \.REMOTE\<0>
010E0007 000F0 P.ABC: .LONG 17694727
00000000 000F4 .ADDRESS P.ABD
00 00 48 43 54 41 42 2E 000F8 P.ABF: .ASCII \.BATCH\<0><0>
010E0006 00100 P.ABE: .LONG 17694726
00000000 00104 .ADDRESS P.ABF
00 53 53 45 43 4F 52 50 42 55 53 2E 00108 P.ABH: .ASCII \.SUBPROCESS\<0>
010E000B 00114 P.ABG: .LONG 17694731
00000000 00118 .ADDRESS P.ABH
45 52 55 4C 49 41 46 2E 0011C P.ABJ: .ASCII \.FAILURE\
010E0008 00124 P.ABI: .LONG 17694728
00000000 00128 .ADDRESS P.ABJ
53 53 45 43 43 55 53 2E 0012C P.ABL: .ASCII \.SUCCESS\
010E0008 00134 P.ABK: .LONG 17694728
00000000 00138 .ADDRESS P.ABL
00 56 52 50 53 59 53 2E 0013C P.ABN: .ASCII \.SYSPRV\<0>
010E0007 00144 P.ABM: .LONG 17694727
00000000 00148 .ADDRESS P.ABN
00 53 53 41 50 59 42 2E 0014C P.ABP: .ASCII \.BYPASS\<0>
010E0007 00154 P.ABO: .LONG 17694727
00000000 00158 .ADDRESS P.ABP
45 44 41 52 47 50 55 2E 0015C P.ABR: .ASCII \.UPGRADE\
010E0008 00164 P.ABQ: .LONG 17694728
00000000 00168 .ADDRESS P.ABR
00 00 45 44 41 52 47 4E 57 4F 44 2E 0016C P.ABT: .ASCII \.DOWNGRADE\<0><0>
010E000A 00178 P.ABS: .LONG 17694730
00000000 0017C .ADDRESS P.ABT
00 56 52 50 50 52 47 2E 00180 P.ABV: .ASCII \.GRPPRV\<0>
010E0007 00188 P.ABU: .LONG 17694727
00000000 0018C .ADDRESS P.ABV
```


.PSECT SOWNS,NOEXE,2

.PSECT SCODES,NOWRT,2

```

.ENTRY SETSAUDIT, Save R2,R3,R4,R5,R6
MOVAB  OPTION STRING, R6
MOVAB  -96(SPT, SP
CLRB   FLAGS

```

: 0224
:
:
:
: 0225

SETAUDIT
V04-000

H 7
16-Sep-1984 00:41:27
14-Sep-1984 12:08:59

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETAUDIT.B32;1

Page 12
(5)

00	00000000G	00	06	E0	0000F	BBS	#6, CTL\$GQ PROCPRIV+4, 1\$	:	0256	
		7E	2934	8F	3C 00017	MOVZWL	#10548, -(SP)	:	0259	
	00000000G	00		01	FB 0001C	CALLS	#1, LIB\$SIGNAL	:		
					04 00023	RET		:	0258	
0050	8F	00		00	2C 00024	1\$:	MOVCS	#0, (SP), #0, #80, MASK	:	0285
			04	AE	0002B			:		
				52	D4 0002D	CLRL	I	:	0287	
	00000000G	00	6642	DD	0002F	2\$:	PUSHL	OPTION_STRING[I]	:	0289
		16		01	FB 00032	CALLS	#1, CLISPRESENT	:		
00		6E		50	E9 00039	BLBC	R0, 4\$	:		
50		52		52	E2 0003C	BBSS	I, FLAGS, 3\$	:	0292	
				28	C5 00040	3\$:	MULL3	#40, I, R0	:	0294
			04	AE40	9F 00044	PUSHAB	MASK[R0]	:		
				6642	DD 00048	PUSHL	OPTION_STRING[I]	:		
	00000000V	EF		02	FB 0004B	CALLS	#2, GET_VALUES	:		
D9		52		01	F3 00052	4\$:	AOBLEQ	#1, I, 2\$	:	0287
	54	AE		02	D0 00056	MOVL	#2, ARGLIST	:	0305	
	58	AE	04	AE	9E 0005A	MOVAB	MASK, ARGLIST+4	:	0306	
	5C	AE		6E	9E 0005F	MOVAB	FLAGS, ARGLIST+8	:	0307	
			54	AE	9F 00063	PUSHAB	ARGLIST	:	0308	
				EF	9F 00066	PUSHAB	SET_BITS	:		
	00000000G	00	00000000V	02	FB 0006C	CALLS	#2, -SYSS\$CMKRNL	:		
				04	00073	RET		:	0371	

; Routine Size: 116 bytes, Routine Base: \$CODE\$ + 0000


```
377 0372 1 ROUTINE get_values (option, mask) : NOVALUE =
378 0373 BEGIN
379 0374
380 0375 !+++
381 0376
382 0377 Given an ASCII descriptor and a set of masks, decipher the
383 0378 string into an audit class, and set the appropriate bits in the
384 0379 masks
385 0380
386 0381 Inputs:
387 0382     mask - address of mask area
388 0383     option - address of ASCII descriptor
389 0384
390 0385 Outputs:
391 0386     mask - will have a bit set.
392 0387
393 0388 !---
394 0389
395 0390 MAP
396 0391     mask : REF $BBLOCK;                                ! The mask is a byte block
397 0392
398 0393 BIND                                                    ! Locate stuff within the mask:
399 0394     sys = mask[nsa$l_evt_sys],                          ! The simple system bits,
400 0395     loginout = mask[nsa$b_evt_logb] : VECTOR[4,BYTE],    ! Loginout bits
401 0396     file_access = mask[nsa$l_evt_failure] : VECTOR;      ! File access vector
402 0397
403 0398 LOCAL
404 0399     all      : BYTE INITIAL (0),
405 0400     file_all : BYTE INITIAL (0),
406 0401     string   : $BBLOCK[dsc$c_s_bln],
407 0402     desc     : $BBLOCK[dsc$c_s_bln];
408 0403
409 0404 $init_dyndesc(desc);                                ! Get a couple of
410 0405 $init_dyndesc(string);                              ! dynamic descriptors
411 0406
412 0407 str$concat(desc, .option, %ASCIIID '.ALL');
413 0408 IF cli$present(desc) THEN all = 1;
414 0409
415 0410 !
416 0411 ! system events
417 0412 !
418 0413
419 0414 INCR i FROM 0 TO num_sys_events-1 DO
420 0415 BEGIN
421 0416     str$concat(desc, .option, .sys_events[i]);
422 0417     IF .all OR cli$present(desc) THEN sys = .sys OR .sys_bits[i];
423 0418 END;
424 0419
425 0420 !
426 0421 ! Loginout events
427 0422 !
428 0423
429 0424 INCR i FROM 0 TO num_loginout_events-1 DO
430 0425 BEGIN
431 0426     str$concat(desc, .option, .loginout_events[i]);
432 0427     IF .all OR cli$present(desc)
433 0428 THEN
```

```
434 0429 4 BEGIN
435 0430 4 INCR j FROM 0 TO num_loginout_types-1 DO
436 0431 5 BEGIN
437 0432 5 IF (.i EQLU 0) AND (.j EQLU 6) THEN EXITLOOP;
438 0433 5 str$concat(desc, .option, .loginout_events[i], .loginout_types[j]);
439 0434 5 IF .all OR cli$present(desc)
440 0435 5 THEN
441 0436 5 loginout[i] = .loginout[i] OR .loginout_bits[j];
442 0437 5 END;
443 0438 4 END;
444 0439 2 END;
445 0440 2 loginout[0] = .loginout[0] AND NOT (nsa$m_evt_log_bat OR nsa$m_evt_log_sub);
446 0441 2
447 0442 2 file access events
448 0443 2
449 0444 2
450 0445 2 str$concat(desc, .option, %ASCID '.FILE_ACCESS', %ASCID '.ALL');
451 0446 2 IF cli$present(desc) THEN file_all = 1;
452 0447 2
453 0448 2 INCR i FROM 0 TO num_file_events-1 DO
454 0449 3 BEGIN
455 0450 3 str$concat(desc, .option, %ASCID '.FILE_ACCESS', .file_events[i]);
456 0451 3 IF .all OR .file_all OR cli$present(desc)
457 0452 3 THEN
458 0453 4 BEGIN
459 0454 4 INCR j FROM 0 TO num_access_types-1 DO
460 0455 5 BEGIN
461 0456 5 str$concat(desc, .option, %ASCID '.FILE_ACCESS', .file_events[i], .access_types[j]);
462 0457 5 IF .all OR .file_all OR cli$present(desc)
463 0458 5 THEN
464 0459 5 file_access[i] = .file_access[i] OR .access_bits[j];
465 0460 5 END;
466 0461 4 IF .file_access[i] EQLU 0 THEN file_access[i] = .access_bits[0];
467 0462 4 END;
468 0463 3 END;
469 0464 2
470 0465 2 Upgrade and Downgrade not implemented yet so make sure that ALL has not
471 0466 2 turned them on.
472 0467 2
473 0468 2 file_access[4] = 0;
474 0469 2 file_access[5] = 0;
475 0470 2
476 0471 1 END;
```

```
53 53 45 43 43 41 5F 45 4C 4C 41 2E 001FC P.ACL: .ASCII \.ALL\
010E0004 00200 P.ACK: .LONG 17694724
00000000 00204 .ADDRESS P.ACL
4C 49 46 2E 00208 P.ACN: .ASCII \.FILE_ACCESS\
010E000C 00214 P.ACM: .LONG 17694732
00000000 00218 .ADDRESS P.ACN
4C 4C 41 2E 0021C P.ACP: .ASCII \.ALL\
010E0004 00220 P.ACO: .LONG 17694724
00000000 00224 .ADDRESS P.ACP
```

.PSECT \$SPLITS,NOWRT,NOEXE,2

[illegible]

.PSECT \$CODES,NOWRT,2

Address	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	
---------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

	06	52	D1	0009A	CMPL	J, #6	
		26	13	0009D	BEQL	11\$	
	10	A942	DD	0009F	8\$: PUSH	LOGINOUT_TYPES[J]	0433
		6943	DD	000A3	PUSHL	LOGINOUT_EVENTS[I]	
		56	DD	000A6	PUSHL	R6	
	0C	AE	9F	000AB	PUSHAB	DESC	
6A		04	FB	000AB	CALLS	#4, STR\$CONCAT	
08		58	E8	000AE	BLBS	ALL, 9\$	0434
		5E	DD	000B1	PUSHL	SP	
6B		01	FB	000B3	CALLS	#1, CL\$PRESENT	
08		50	E9	000B6	BLBC	R0, 10\$	
	30	A942	DF	000B9	9\$: PUSH	LOGINOUT_BITS[J]	0436
6344		9E	88	000BD	BISB2	@(SP)+, (I)[R4]	
52		07	F3	000C1	10\$: AOBLEQ	#7, J, 7\$	0430
53		03	F3	000C5	11\$: AOBLEQ	#3, I, 5\$	0424
64		21	8A	000C9	BICB2	#33, (R4)	0440
		EF	9F	000CC	PUSHAB	P.ACO	0446
		EF	9F	000D2	PUSHAB	P.ACM	
		56	DD	000D8	PUSHL	R6	
	0C	AE	9F	000DA	PUSHAB	DESC	
6A		04	FB	000DD	CALLS	#4, STR\$CONCAT	
		5E	DD	000E0	PUSHL	SP	0447
6B		01	FB	000E2	CALLS	#1, CL\$PRESENT	
03		50	E9	000E5	BLBC	R0, 12\$	
57		01	90	000E8	MOVB	#1, FILE_ALL	
		52	D4	000EB	12\$: CLRL	I	0449
	50	A942	DD	000ED	13\$: PUSH	FILE_EVENTS[I]	0451
		EF	9F	000F1	PUSHAB	P.ACO	
		56	DD	000F7	PUSHL	R6	
	0C	AE	9F	000F9	PUSHAB	DESC	
6A		04	FB	000FC	CALLS	#4, STR\$CONCAT	
0B		58	E8	000FF	BLBS	ALL, 14\$	0452
08		57	E8	00102	BLBS	FILE_ALL, 14\$	
		5E	DD	00105	PUSHL	SP	
6B		01	FB	00107	CALLS	#1, CL\$PRESENT	
3C		50	E9	0010A	BLBC	R0, 18\$	
		53	D4	0010D	14\$: CLRL	J	0455
	70	A943	DD	0010F	15\$: PUSH	ACCESS_TYPES[J]	0457
		50	A942	DD	00113	PUSHL	FILE_EVENTS[I]
		EF	9F	00117	PUSHAB	P.ACS	
		56	DD	0011D	PUSHL	R6	
	10	AE	9F	0011F	PUSHAB	DESC	
6A		05	FB	00122	CALLS	#5, STR\$CONCAT	
0B		58	E8	00125	BLBS	ALL, 16\$	0458
08		57	E8	00128	BLBS	FILE_ALL, 16\$	
		5E	DD	0012B	PUSHL	SP	
6B		01	FB	0012D	CALLS	#1, CL\$PRESENT	
07		50	E9	00130	BLBC	R0, 17\$	
6542		0088	C943	C8	00133	16\$: BISL2	ACCESS_BITS[J], (R5)[I]
53		05	F3	0013A	17\$: AOBLEQ	#5, J, 15\$	0455
		6542	D5	0013E	TSTL	(R5)[I]	0462
		06	12	00141	BNEQ	18\$	
6542		0088	C9	D0	00143	MOVL	ACCESS_BITS, (R5)[I]
52		07	F3	00149	18\$: AOBLEQ	#7, I, 13\$	0449
	10	A5	7C	0014D	CLRQ	16(R5)	0469
		04	00150	RET			0471

SETAUDIT
V04-000

M 7
16-Sep-1984 00:41:27
14-Sep-1984 12:08:59

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETAUDIT.B32;1

Page 17
(6)

; Routine Size: 337 bytes, Routine Base: \$CODE\$ + 0074

```

478 0472 1 ROUTINE set_bits (mask, flags) : NOVALUE =
479 0473 BEGIN
480 0474
481 0475 !+++
482 0476
483 0477 Reset the bits for either security alarms, or security journaling.
484 0478
485 0479 Inputs:
486 0480     mask - address of vector of bits to enable and disable
487 0481     flags - address of flags (alarm, journal)
488 0482
489 0483 Outputs:
490 0484     None. The auditing bits are set.
491 0485
492 0486 !---
493 0487
494 0488 MAP
495 0489     mask : REF VECTOR[.BYTE],
496 0490     flags : REF BITVECTOR;
497 0491
498 0492 BIND
499 0493     enab = mask[0] : VECTOR[.BYTE],
500 0494     disab = mask[nsa$sk_evt_length] : VECTOR[.BYTE];
501 0495
502 0496 LOCAL
503 0497     bits : REF VECTOR[.BYTE];
504 0498
505 0499
506 0500 Determine whether to modify the journal or alarm bits.
507 0501
508 0502 IF .flags[v_alarm]
509 0503 THEN bits = nsa$gr_alarmvec
510 0504 ELSE bits = nsa$gr_journvec;
511 0505 bits = nsa$gr_alarmvec;
512 0506
513 0507
514 0508 Step through the vector in SYSCOMMON and enable bits.
515 0509
516 0510 INCR i FROM 0 TO nsa$sk_evt_length-1 DO
517 0511     bits[i] = (.bits[i] OR .enab[i]) AND NOT .disab[i];
518 0512
519 0513 RETURN;
520 0514 1 END;
```

001C 00000 SET_BITS:

53	04	AC	28	C1 00002	.WORD	Save R2,R3,R4	0472
		51 00000000G	00	9E 00007	ADDL3	#40, MASK, R3	0494
			50	D4 0000E	MOVAB	NSA\$GR_ALARMVEC, BITS	0505
		52	6041	9A 00010	CLRL	1	0511
		54	04 BC40	9A 00014	MOVZBL	(1)[BITS], R2	
		52	54	C8 00019	MOVZBL	@MASK[1], R4	
6041		52	6043	8B 0001C	BISL2	R4, R2	
					BICB3	(1)[R3], R2, (1)[BITS]	

SETAUDIT
V04-000

B 8
16-Sep-1984 00:41:27
14-Sep-1984 12:08:59

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETAUDIT.B32;1

Page 19
(7)

EA

50

27 F3 00022
04 00026

AOBLEQ #39, I, 1\$
RET

: 0514

; Routine Size: 39 bytes, Routine Base: \$CODE\$ + 01C5

```
: 521      0515 1 |
: 522      0516 1 |ROUTINE start_journal (chan) =
: 523      0517 1 |BEGIN
: 524      0518 1 |
: 525      0519 1 |+++
: 526      0520 1 |
: 527      0521 1 |Attempt to assign a channel to the journal.
: 528      0522 1 |
: 529      0523 1 |Inputs:
: 530      0524 1 |    chan - address of word to store channel number
: 531      0525 1 |
: 532      0526 1 |Outputs:
: 533      0527 1 |    chan - will be filled in.
: 534      0528 1 |
: 535      0529 1 |---
: 536      0530 1 |
: 537      0531 1 |MAP
: 538      0532 1 |    chan : REF VECTOR[WORD];
: 539      0533 1 |
: 540      0534 1 |RETURN $ASSJNL(CHAN = chan[0],
: 541      0535 1 |                JNL TYP = dt$_atjn[,
: 542      0536 1 |                ACMODE = UPLIT BYTE(jsb$c_exec),
: 543      0537 1 |                JNLNAM = %ASCII 'SECURITY');
: 544      0538 1 |
: 545      0539 1 |END;
```

! Try to assign a channel,
! perhaps causing the journal
! to get started on the
! system.


```
546 0540 1 |
547 0541 1 |ROUTINE create_journal (chan, disk) =
548 0542 1 |BEGIN
549 0543 1 |
550 0544 1 |+++
551 0545 1 |
552 0546 1 |Try to create a journal.
553 0547 1 |
554 0548 1 |Inputs:
555 0549 1 |    chan - address of word, where to put channel number
556 0550 1 |    disk - address of descriptor for device to journal to
557 0551 1 |
558 0552 1 |Outputs:
559 0553 1 |    chan - will be filled in.
560 0554 1 |
561 0555 1 |---
562 0556 1 |
563 0557 1 |MAP
564 0558 1 |    chan : REF VECTOR[WORD],
565 0559 1 |    disk : REF $BBLOCK;
566 0560 1 |
567 0561 1 |LOCAL
568 0562 1 |    device_list : VECTOR[3],
569 0563 1 |    data : $BBLOCK[jsb$length];
570 0564 1 |
571 0565 1 |
572 0566 1 |Set up the list of descriptors pointing to the device(s) on which the
573 0567 1 |journal file resides.
574 0568 1 |
575 0569 1 |IF .disk[dsc$length] NEQ 0
576 0570 1 |THEN
577 0571 1 |    BEGIN
578 0572 1 |        device_list[0] = .disk[dsc$length];
579 0573 1 |        device_list[1] = .disk[dsc$a_pointer];
580 0574 1 |    END
581 0575 1 |ELSE
582 0576 1 |    BEGIN
583 0577 1 |        device_list[0] = %CHARCOUNT('SYSSYSDEVICE');
584 0578 1 |        device_list[1] = UPLIT BYTE ('SYSSYSDEVICE');
585 0579 1 |    END;
586 0580 1 |device_list[2] = 0;
587 0581 1 |
588 0582 1 |
589 0583 1 |
590 0584 1 |Set up the Journal Specification Block, which contains all the
591 0585 1 |information about the journal to be opened/created.
592 0586 1 |
593 0587 1 |CHSFILL(0, jsb$length, data);
594 0588 1 |data[jsb$w_inlnamlen] = %CHARCOUNT('SECURITY');
595 0589 1 |data[jsb$l_inlnam] = UPLIT BYTE('SECURITY');
596 0590 1 |data[jsb$b_inltyp] = jsb$at;
597 0591 1 |data[jsb$b_inldev] = jsb$disk;
598 0592 1 |data[jsb$w_maxsiz] = nsa$ss_idt_record_buf;
599 0593 1 |data[jsb$w_filext] = 10;
600 0594 1 |data[jsb$w_bufsiz] = 1;
601 0595 1 |data[jsb$b_acmode] = jsb$exec;
602 0596 1 |data[jsb$w_prot] = %B'1101T10111001100';
```

! If there's a valid device specified by the user, transfer the length and address to DEVICE_LIST.

! Otherwise, use SYSSYSDEVICE.

! Zero-terminate the list

! Initialize Journal name is 8 bytes, name is "SECURITY"

! AT journal always a disk journal and can be this big extend size

! (Do this to stop crashing)

! do everything from EXEC

! Protection=(S:RW,O:RW,G:W,W:W)

```

: 603      0597 1 !data[jsb$l_uic] = %X'00010004';
: 604      0598 1 !data[jsb$l_flags] = jsb$m_known
: 605      0599 1 !data[jsb$l_flags] = jsb$m_known OR jsb$m_cif;
: 606      0600 1 !data[jsb$b_copies] = 1;
: 607      0601 1 !data[jsb$l_prinamdes] = device_list;
: 608      0602 1
: 609      0603 1 !RETURN $CREJNL(CHAN = chan[0],
: 610      0604 1 !          FLAGS = cifs$m_read,
: 611      0605 1 !          ACMODE = UPLIT BYTE(jsb$sc_exec),
: 612      0606 1 !          JSB = data);
: 613      0607 1
: 614      0608 1 !END;

```

! Owner = [1,4]
! Known journal, and create a
! journal file if not there
! Only one copy of the file
! Disk names here
! Give it a try.


```

615 0609 1 |
616 0610 1 |ROUTINE new_version (chan, disk) =
617 0611 1 |BEGIN
618 0612 1 |
619 0613 1 |+++
620 0614 1 |
621 0615 1 |Create a new version of the journal file.
622 0616 1 |
623 0617 1 |Inputs:
624 0618 1 |    chan - address of channel to use
625 0619 1 |    disk - address of device descriptor, telling what device to
626 0620 1 |           locate the new file.
627 0621 1 |
628 0622 1 |Outputs:
629 0623 1 |    None.
630 0624 1 |
631 0625 1 |---
632 0626 1 |
633 0627 1 |MAP
634 0628 1 |    chan : REF VECTOR[WORD],
635 0629 1 |    disk : REF $BBLOCK;
636 0630 1 |
637 0631 1 |LOCAL
638 0632 1 |    item_list : VECTOR[12],
639 0633 1 |    curdev_len,
640 0634 1 |    curdev : VECTOR[10];
641 0635 1 |
642 0636 1 |
643 0637 1 |Get the current device name.
644 0638 1 |
645 0639 1 |BEGIN
646 0640 1 |LOCAL
647 0641 1 |    status,
648 0642 1 |    list : $ITMLST_DECL(ITEMS = 1);
649 0643 1 |
650 0644 1 |$ITMLST_INIT(ITMLST = list,
651 0645 1 |    (ITMCO = cji$_fildsknam,
652 0646 1 |    BUFA = curdev,
653 0647 1 |    BUFSIZ = %ALLOCATION(curdev),
654 0648 1 |    RETLEN = curdev_len)
655 0649 1 |);
656 0650 1 |IF NOT (status = $GETCJI(ITMLST = list,
657 0651 1 |    CHAN = .chan[0]))
658 0652 1 |THEN RETURN .status;
659 0653 1 |END;
660 0654 1 |
661 0655 1 |
662 0656 1 |
663 0657 1 |The $CRENWV service takes as its argument a list of itemlist addresses.
664 0658 1 |Set that up now.
665 0659 1 |
666 0660 1 |item_list[0] = item_list[2];
667 0661 1 |item_list[1] = 0;
668 0662 1 |
669 0663 1 |
670 0664 1 |The journaling facility uses a pseudo-itemlist, which actually looks more
671 0665 1 |like a descriptor list. In the first longword is the itemcode and the

```

```

! Declare these here,
! because they're temporary.

```

```

! Set up this list,
! want device name
! store it here,
! up to this big,
! actual size returned here.

```

```

! Do it.
! If a problem,
! stop now.

```

```

! Point to "real" itemlist
! Zero-terminate.

```

```
: 672      0666 1 ! length of the item; the second longword contains the address of the item.
: 673      0667 1 ! This list must be zero-terminated as well. In all cases, we want to list
: 674      0668 1 ! the current device name, and the "close this file" flag.
: 675      0669 1
: 676      0670 1 ! item_list[2] = (cnv$_flags^16) OR 4;          ! Flags are a longword
: 677      0671 1 ! item_list[3] = UPLIT(cnv$m_close);          ! and say "close the file"
: 678      0672 1
: 679      0673 1 ! item_list[4] = (cnv$_curdevnam^16) OR .curdev_len;    ! Current devname
: 680      0674 1 ! item_list[5] = curdev;
: 681      0675 1
: 682      0676 1
: 683      0677 1 ! If the user wants the journal file on a different device, then put that
: 684      0678 1 ! item in the list, followed by the zero terminator.
: 685      0679 1
: 686      0680 1 ! IF .disk[dsc$_length] NEQ 0          ! If to a different disk,
: 687      0681 1 ! THEN                                ! put that item in.
: 688      0682 1 !     BEGIN
: 689      0683 1 !         item_list[6] = (cnv$_newdevnam^16) OR .disk[dsc$_length];
: 690      0684 1 !         item_list[7] = .disk[dsc$_pointer];
: 691      0685 1 !         item_list[8] = 0;
: 692      0686 1 !     END
: 693      0687 1
: 694      0688 1
: 695      0689 1 ! Otherwise, simply zero-terminate the list.
: 696      0690 1
: 697      0691 1 ! ELSE item_list[6] = 0;
: 698      0692 1
: 699      0693 1 ! RETURN $CRENVV(CHAN = .chan[0],          ! Do it.
: 700      0694 1 !                     LSTADR = item_list);
: 701      0695 1
: 702      0696 1 ! END;
```


SETAUDIT
V04-000

H 8
16-Sep-1984 00:41:27
14-Sep-1984 12:08:59

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETAUDIT.B32;1

Page 25
(11)

: 703 0697 1 !
: 704 0698 1 END
: 705 0699 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
SPLITS	592	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
SOWNS	192	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
SCODES	492	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	33	0	1000	00:01.8

COMMAND QUALIFIERS

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

: Size: 492 code + 784 data bytes
: Run Time: 00:15.4
: Elapsed Time: 00:49.9
: Lines/CPU Min: 2723
: Lexemes/CPU-Min: 14758
: Memory Used: 160 pages
: Compilation Complete

0052 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

