

CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL

```

SSSSSSSS EEEEEEEEE TTTTTTTTT DDDDDDD IIIII RRRRRRR
SSSSSSSS EEEEEEEEE TTTTTTTTT DDDDDDD IIIII RRRRRRR
SS          EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
SS          EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
SS          EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SSSSSS      EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SSSSSS      EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SS          EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SS          EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SS          EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SSSSSSS      EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR
          SSSSSSS      EEEEEEEEE TTTTTTTTT DD      DD I      RR      RR

```

```

LL          IIIII SSSSSSS
LL          IIIII SSSSSSS
LL          I      SS
LL          I      SS
LL          I      SS
LL          I      SS
LL          I      SSSSSS
LL          I      SSSSSS
LL          I      SS
LL          I      SS
LL          I      SS
LL          I      SS
LLLLLLLLLL IIIII SSSSSSS
LLLLLLLLLL IIIII SSSSSSS

```

```

1 0001 0 M DULE setdir (
2 0002 0 IDENT = 'V04-000',
3 0003 0 ADDRESSING_MODE(EXTERNAL=GENERAL,
4 0004 0 NONEXTERNAL=LONG_RELATIVE)
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
13 0013 1 * ALL RIGHTS RESERVED. *
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
20 0020 1 * TRANSFERRED. *
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
24 0024 1 * CORPORATION. *
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
28 0028 1 *
29 0029 1 *
30 0030 1 *****
31 0031 1
32 0032 1
33 0033 1 +-
34 0034 1 FACILITY: Set Directory Command
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1
38 0038 1 This module processes the Set Directory command.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 Vax native, privileged user mode
43 0043 1
44 0044 1 --
45 0045 1
46 0046 1 AUTHOR: Gerry Smith CREATION DATE: 3-Nov-1981
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1 V03-007 AEW0001 Anne E. Warner 6-Mar-1984
51 0051 1 Add error messages to prevent the use of search lists
52 0052 1 with devices for directories.
53 0053 1 FILSPCSRCH - File specification contains a searchlist
54 0054 1 NOSRCHLST - Command does not support search lists
55 0055 1 Both are defined in SHRMSG.MSG.
56 0056 1
57 0057 1 V03-006 GAS0170 Gerry Smith 23-Aug-1983

```

```

: 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 |
: 70      0070 1 |
: 71      0071 1 |
: 72      0072 1 |
: 73      0073 1 |
: 74      0074 1 |
: 75      0075 1 |
: 76      0076 1 |
: 77      0077 1 |
: 78      0078 1 |
: 79      0079 1 |
: 80      0080 1 |
: 81      0081 1 |**

```

Instead of simply doing an IO\$\_ACCESS to get the directory's header, perform an actual open, by issuing an IO\$\_ACCESS OR IO\$\_M\_ACCESS, to insure proper file locking by the XQP.

V03-005 GAS0116 Gerry Smith 6-Apr-1983  
Use the common qualifier routines.

V03-004 GAS0112 Gerry Smith 29-Mar-1983  
Fix for the new CLI interface, as well as the new command dispatcher.

V03-003 GAS0091 Gerry Smith 19-Oct-1982  
Change input request for new CLD syntax.

V03-002 GAS0026 Gerry Smith 18-Dec-1981  
Use shared message file, and lower fatal messages to simple error messages.

V03-001 GAS0024 Gerry Smith 14-Dec-1981  
Fix /LOG logic

SETDIR  
V04-000

J 16  
16-Sep-1984 00:59:34  
14-Sep-1984 12:09:06

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

Page 3  
(2)

```
: 83      0082 1 LIBRARY 'SYS$LIBRARY:LIB';  
: 84      0083 1 LIBRARY 'SYS$LIBRARY:TPAMAC';  
: 85      0084 1
```

```

: 87      0085 1 FORWARD ROUTINE
: 88      0086 1   set$directory : NOVALUE,      ! Main routine for directory
: 89      0087 1   get_qual,           ! Get qualifiers and values
: 90      0088 1   set_attributes,       ! Routine to set directory attributes
: 91      0089 1   get_dir;           ! Routine to get each directory
: 92      0090 1
: 93      0091 1
: 94      0092 1 EXTERNAL ROUTINE
: 95      0093 1   cli$present,         ! Determine presence of qualifier
: 96      0094 1   cli$get_value,       ! Get value of qualifier
: 97      0095 1   lib$qual_file_parse, ! Get common qualifiers
: 98      0096 1   lib$qual_file_match, ! See if directory matches common quals
: 99      0097 1   lib$file_scan,       ! Routine to get next directory
100     0098 1   check_privilege : NOVALUE, ! Routine to check for privilege
101     0099 1   search_error,         ! Where to go if file search fails
102     0100 1   file_error,          ! Where to go if file error occurs
103     0101 1   parse_uic,           ! Routine for parsing UIC strings
104     0102 1   lib$tparse,          ! Parser
105     0103 1   lib$cvd_dtb,         ! Convert decimal to number
106     0104 1   sys$fao;           ! Formatted ASCII output
107     0105 1
108     0106 1 !
109     0107 1 ! Declare external data
110     0108 1 !
111     0109 1 EXTERNAL
112     0110 1   setfile$flags : BITVECTOR[32], ! Qualifier flags word
113     0111 1   uic_value,           ! Owner UIC
114     0112 1   group,              ! UIC group number
115     0113 1   member,             ! UIC member number
116     0114 1   vrsn_value,         ! Version limit
117     0115 1 !
118     0116 1 ! RMS structures
119     0117 1 !
120     0118 1   file_result : VECTOR[nam$c_maxrss, BYTE], ! Resultant name string
121     0119 1   file_expanded : VECTOR[nam$c_maxrss, BYTE], ! Expanded name string
122     0120 1 ! file r[f is in own section because it not externally defined.
123     0121 1   file_nam : $BLOCK[nam$c_bln], ! File name block
124     0122 1   file_fab : $BLOCK[fab$c_bln]; ! File FAB
125     0123 1 !
126     0124 1 !
127     0125 1 ! Error messages
128     0126 1 !
129     0127 1 EXTERNAL LITERAL
130     0128 1   lib$_filfaimat,       ! No match for common quals
131     0129 1   lib$_quipro,         ! Quit processing
132     0130 1   set$_facility,        ! Facility code
133     0131 1   set$_closeerr,       ! Could not close file
134     0132 1   set$_entered,        ! File entered in a directory
135     0133 1   set$_enterr,         ! Error entering file
136     0134 1   set$_modified,      ! File/directory modified
137     0135 1   set$_notdir,        ! Not a directory
138     0136 1   set$_notods2,       ! Not an ODS2 structure
139     0137 1   set$_readerr,       ! Could not read file
140     0138 1   set$_remerr,        ! Could not remove file
141     0139 1   set$_removed,       ! Directory entry removed
142     0140 1   set$_writeerr;      ! Could not write to file
143     0141 1

```

```

: 144      0142  1  |
: 145      0143  1  |  | Declare some shared messages
: 146      0144  1  |
: 147      P 0145  1  | $SHR_MSGDEF (SET,119,LOCAL,
: 148      P 0146  1  |             (syntax,      error),
: 149      P 0147  1  |             (openin,     error),
: 150      P 0148  1  |             (valerr,     error),
: 151      P 0149  1  |             (filspcsrc,  error),
: 152      0150  1  |             (nosrchlst,  error));
: 153      0151  1  |
: 154      0152  1  |
: 155      0153  1  |  |
: 156      0154  1  |  | Literal data definitions
: 157      0155  1  |  |
: 158      0156  1  | LITERAL
: 159      0157  1  |     true = 1
: 160      0158  1  |     false = 0;
: 161      0159  1  |
: 162      0160  1  | LITERAL
: 163      P 0161  1  |     SEQULST
: 164      P 0162  1  |         (QUAL_,,1,1,
: 165      P 0163  1  |         (quit,);
: 166      P 0164  1  |         (log,);
: 167      P 0165  1  |         (owner,);
: 168      0166  1  |         (vrsn,));
: 169      0167  1  |
: 170      0168  1  |
: 171      0169  1  |  | Declare the context longword
: 172      0170  1  |  |
: 173      0171  1  |  | OWN
: 174      0172  1  |     file_rlf : $NAM(),
: 175      0173  1  |     context;
: 176      0174  1  |

```

```

: quit processing
: LOG bit
: OWNER UIC bit
: VERSION_LIMIT bit

```

SETDIR  
V04-000

M 16  
16-Sep-1984 00:59:34  
14-Sep-1984 12:09:06

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

Page 6  
(4)

```

: 178      0175 1 |
: 179      0176 1 | TPARSE table for directories
: 180      0177 1 |
: 181      0178 1 $INIT_STATE (dir_state,dir_keys);
: 182      0179 1 |
: 183      P 0180 1 $STATE (dirstart,
: 184      0181 1 | (tpa$_octal,...,group));           ! Get the group number
: 185      0182 1 |
: 186      P 0183 1 $STATE (
: 187      0184 1 | (','));                           ! Get rid of the comma
: 188      0185 1 |
: 189      P 0186 1 $STATE (
: 190      0187 1 | (tpa$_octal,tpa$_exit,...,member)); ! Get the member number
: 191      0188 1 |
: 192      0189 1 PSECT OWN = $OWNS;
: 193      0190 1 PSECT GLOBAL = $GLOBALS;
```



```

195 0191 1 GLOBAL ROUTINE set$directory : NOVALUE =
196 0192 1 **
197 0193 1
198 0194 1 Functional description
199 0195 1
200 0196 1 This is the main control module for SET DIRECTORY. It calls
201 0197 1 LIB$FILE_SCAN to perform the necessary functions on the directory
202 0198 1 specified in the call to SET.
203 0199 1
204 0200 1 Calling sequence
205 0201 1
206 0202 1 CALL set$directory()
207 0203 1
208 0204 1 Input parameters
209 0205 1 none
210 0206 1
211 0207 1 Output parameters
212 0208 1 none
213 0209 1
214 0210 1 Implicit outputs
215 0211 1 none
216 0212 1
217 0213 1 Routine value
218 0214 1 none
219 0215 1
220 0216 1 Side effects
221 0217 1 none
222 0218 1
223 0219 1 --
224 0220 2 BEGIN
225 0221 2
226 0222 2
227 0223 2 LOCAL
228 0224 2 status,
229 0225 2 dir_desc : $BLOCK[dsc$c_s_bln];
230 0226 2
231 0227 2
232 0228 2 Check that the image is running with appropriate privilege.
233 0229 2
234 0230 2 check_privilege();
235 0231 2
236 0232 2
237 0233 2 Get the common qualifiers.
238 0234 2
239 0235 2 status = lib$qual_file_parse(%REF(lib$m_cqf_exclude OR
240 0236 2 lib$m_cqf_before OR
241 0237 2 lib$m_cqf_since OR
242 0238 2 lib$m_cqf_created OR
243 0239 2 lib$m_cqf_modified OR
244 0240 2 lib$m_cqf_expired OR
245 0241 2 lib$m_cqf_backup OR
246 0242 2 lib$m_cqf_confirm OR
247 0243 2 lib$m_cqf_byowner),
248 0244 2 context);
249 0245 2 IF NOT .status
250 0246 2 THEN (SIGNAL(.status); RETURN);
251 0247 2

```

```

252 0248 2 1 Get the command qualifiers.
253 0249 2 1
254 0250 2 1
255 0251 2 1 IF NOT get_qual()
256 0252 2 1 THEN RETURN;
257 0253 2 1
258 0254 2 1
259 0255 2 1 For each directory specified, find the directory and perform the
260 0256 2 1 operations requested.
261 0257 2 1
262 0258 2 1 $init_dyndesc(dir_desc); ! Make the descriptor dynamic
263 0259 2 1 file_fab[fab$l_ctx] = false; ! Initialize for get_dir routine
264 0260 2 1
265 0261 2 1 WHILE get_dir(dir_desc)
266 0262 2 1 DO
267 0263 2 1 BEGIN
268 0264 2 1 lib$file_scan( ! For each directory found
269 0265 2 1 file_fab, ! Use this fab
270 0266 2 1 set_attributes, ! Go here if directory found
271 0267 2 1 search_error) ! Or here if error
272 0268 2 1 END;
273 0269 2 1
274 0270 2 1 RETURN;
275 0271 1 1 END;

```

		.TITLE	SETDIR	
		.IDENT	\V04-000\	
		.PSECT	_LIB\$STATES,NOWRT, SHR, PIC,1	
0000	DIR_STATE::			
		.BLKB	0	
0000	DIRSTART:			
		.BLKB	0	
45F4	00000	:TPASTYPE		
		U.2:	.WORD 17908	:
00000000*	00002	:TPASADDR		:
		U.3:	.LONG <<GROUP-U.3>-4>	:
042C	00006	:TPASTYPE		
		U.4:	.WORD 1068	:
55F4	00008	:TPASTYPE		
		U.5:	.WORD 22004	:
00000000*	0000A	:TPASADDR		:
		U.6:	.LONG <<MEMBER-U.6>-4>	:
FFFF	0000E	:TPASTARGET		:
		U.7:	.WORD -1	:
		.PSECT	_LIB\$KEY0\$,NOWRT, SHR, PIC,1	
0000	DIR_KEYS::			
		.BLKB	0	
0000	:TPASKEY0			
	U.1:	.BLKB	0	
		.PSECT	\$OWNS,NOEXE,2	

```

02 00000 FILE_RLF:
      6C 00001 .BYTE 2
      00 00002 .BYTE 96
      00 00003 .BYTE 0
00000000 00004 .LONG 0
      00 00008 .BYTE 0
      00 00009 .BYTE 0
      00 0000A .BYTE 0
      00 0000B .BYTE 0
00000000 0000C .LONG 0
00000000 00010 .LONG 0
      0000# 00014 .WORD 0[8]
      0000# 00024 .WORD 0[3]
      0000# 0002A .WORD 0[3]
00000000 00030 .LONG 0
00000000 00034 .LONG 0
      00 00038 .BYTE 0
      00 00039 .BYTE 0
      00 0003A .BYTE 0
      00 0003B .BYTE 0
      00 0003C .BYTE 0
      00 0003D .BYTE 0
      00# 0003E .BYTE 0[2]
00000000 00040 .LONG 0
00000000 00044 .LONG 0
00000000 00048 .LONG 0
00000000 0004C .LONG 0
00000000 00050 .LONG 0
00000000 00054 .LONG 0
00000000# 00058 .LONG 0[2]
00060 CONTEXT: .BLKB 4

```

```

.EXTRN CLISPRESENT, CLISGET_VALUE
.EXTRN LIB$QUAL_FILE_PARSE
.EXTRN LIB$QUAL_FILE_MATCH
.EXTRN LIB$FILE_SCAN, CHECK_PRIVILEGE
.EXTRN SEARCH_ERROR, FILE_ERROR
.EXTRN PARSE_OIC, LIB$TPARSE
.EXTRN LIB$CVT_DTB, SYSS$FAO
.EXTRN SETFILE$FLAGS, UIC_VALUE
.EXTRN GROUP, MEMBER, VRSN_VALUE
.EXTRN FILE_RESULT, FILE_EXPANDED
.EXTRN FILE_NAM, FILE_FAB
.EXTRN LIB$FILFAIMAT, LIB$QUIPRO
.EXTRN SETS_FACILITY, SETS_CLOSEERR
.EXTRN SETS_ENTERED, SETS_ENTERR
.EXTRN SETS_MODIFIED, SETS_NOTDIR
.EXTRN SETS_NOTODS2, SETS_READERR
.EXTRN SETS_REMERR, SETS_REMOVED
.EXTRN SETS_WRITEERR

```

.PSECT \$CODE\$,NOWRT,2

```

00000000G 5E 0000 00000
00 00 00002
00 FB 00005

```

```

.ENTRY SETSDIRECTORY, Save nothing
SUBL2 #12, SP
CALLS #0, CHECK_PRIVILEGE

```

.....

: 0191  
: 0230

SETDIR  
V04-000

E 1  
16-Sep-1984 00:59:34 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:09:06 [CLIUTL.SRC]SETDIR.B32;1

Page 10  
(5)

04	AE	00000000'	EF	9F	0000C	PUSHAB	CONTEXT	:	0235	
		01FF	8F	3C	00012	MOVZWL	#511, 4(SP)	:	0242	
		04	AE	9F	00018	PUSHAB	4(SP)	:	0235	
00000000G	00		02	FB	0001B	CALLS	#2, LIB\$QUAL_FILE_PARSE	:		
	0A		50	EB	00022	BLBS	STATUS, 1\$	:	0245	
			50	DD	00025	PUSHL	STATUS	:	0246	
00000000G	00		01	FB	00027	CALLS	#1, LIB\$SIGNAL	:		
				04	0002E	RET		:		
00000000V	EF		00	FB	0002F	1\$:	CALLS	#0, GET_QUALS	:	0251
	39		50	E9	00036		BLBC	RO, 3\$	:	
04	AE	020E0000	8F	D0	00039		MOVL	#3471936, DIR_DESC	:	0258
		08	AE	D4	00041		CLRL	DIR_DESC+4	:	
		00000000G	00	D4	00044		CLRL	FILE_FAB+24	:	0259
		04	AE	9F	0004A	2\$:	PUSHAB	DIR_DESC	:	0261
00000000V	EF		01	FB	0004D		CALLS	#1, GET_DIR	:	
	1B		50	E9	00054		BLBC	RO, 3\$	:	
		00000000G	00	9F	00057		PUSHAB	SEARCH_ERROR	:	0264
		00000000V	EF	9F	0005D		PUSHAB	SET_ATTRIBUTES	:	
		00000000G	00	9F	00063		PUSHAB	FILE_FAB	:	
00000000G	00		03	FB	00069		CALLS	#3, LIB\$FILE_SCAN	:	
			D8	11	00070		BRB	2\$	:	0263
			04	00072	3\$:		RET	:	0271	

; Routine Size: 115 bytes, Routine Base: \$CODE\$ + 0000

```

277 0272 1 ROUTINE get_qual =
278 0273 1 :
279 0274 1 :
280 0275 1 : This routine interrogates the CLI for qualifiers and values. If an
281 0276 1 : error occurs, it is signaled and control is returned with a value of
282 0277 1 : FALSE.
283 0278 1 :
284 0279 1 :--
285 0280 2 BEGIN
286 0281 2
287 0282 2 LOCAL
288 0283 2     status,
289 0284 2     desc : $BLOCK[dsc$c_s_bln];           ! General descriptor
290 0285 2
291 0286 2 $init_dyndesc(desc);                   ! Make it dynamic
292 0287 2
293 0288 2 :
294 0289 2 : Get the simple flags first.
295 0290 2
296 0291 2 setfile$flags[qual_log] = cli$present(%ASCID 'LOG');
297 0292 2
298 0293 2 :
299 0294 2 : /OWNER_UIC
300 0295 2
301 0296 2 IF cli$present(%ASCID 'OWNER_UIC')
302 0297 2 THEN
303 0298 2     BEGIN
304 0299 2         setfile$flags[qual_owner] = 1;           ! If present, set bit
305 0300 2
306 0301 3 : If no value on /OWNER_UIC, then call GETJPI to get this process's UIC.
307 0302 3 :
308 0303 3 IF NOT cli$get_value(%ASCID 'OWNER_UIC', desc)
309 0304 3 THEN
310 0305 4     BEGIN
311 0306 4         LOCAL
312 0307 4             iosb : VECTOR[4,WORD];
313 0308 4             status = $GETJPIW(ITMLST = UPLIT(WORD(4,jpi$_uic),
314 0309 4                 uic_value,
315 0310 4                 0,
316 0311 4                 0),
317 0312 4                 IOSB = iosb);
318 0313 4
319 0314 4 IF .status
320 0315 4 THEN status = .iosb[0];
321 0316 4 IF NOT .status
322 0317 5 THEN
323 0318 5     BEGIN
324 0319 5         SIGNAL(.status);
325 0320 4         RETURN false;
326 0321 4     END;
327 0322 4
328 0323 4 : Otherwise, assume a real UIC. Parse the string.
329 0324 4 :
330 0325 3 ELSE
331 0326 4     BEGIN
332 0327 5 IF NOT (status = parse_uic(desc, uic_value))
333 0328 4 THEN

```

```

334 03329 5 BEGIN
335 03330 5 SIGNAL(set$_syntax, 1, desc);
336 03331 5 RETURN false;
337 03332 4 END;
338 03333 4 END;
339 03334 2 END;
340 03335 2
341 03336 2 !
342 03337 2 ! Now for the version limit.
343 03338 2 !
344 03339 2 IF cli$present(%ASCID 'VERSION_LIMIT')
345 03340 2 THEN
346 03341 3 BEGIN
347 03342 3 setfile$flags[qual_vrsn] = 1; ! Show that /VERSION specified
348 03343 3 vrsn_value = 32767; ! Set to the default
349 03344 3 IF cli$get_value(%ASCID 'VERSION_LIMIT', desc)
350 03345 3 THEN
351 03346 4 BEGIN
352 03347 4 IF NOT lib$cvt_dtb(.desc[dsc$_length],
353 03348 4 .desc[dsc$_pointer],
354 03349 4 vrsn_value)
355 03350 4 THEN
356 03351 5 BEGIN
357 03352 5 SIGNAL(set$_syntax, 1, desc);
358 03353 5 RETURN false;
359 03354 4 END;
360 03355 4 IF .vrsn_value LSS 0
361 03356 4 OR .vrsn_value GTR 65535
362 03357 4 THEN (SIGNAL(set$_valerr); RETURN false);
363 03358 3 END;
364 03359 2 END;
365 03360 2
366 03361 2 RETURN true;
367 03362 1 END;

```

											.PSECT \$SPLITS,NOWRT,NOEXE,2								
											00	47	4F	4C	00000	P.AAB:	.ASCII	\LOG\<0>	
															010E0003	00004	P.AAA:	.LONG	17694723
															00000000'	00008		.ADDRESS	P.AAB
00	00	00	43	49	55	5F	52	45	4E	57	4F	0000C	P.AAD:	.ASCII	\OWNER UIC\<0><0><0>				
															010E0009	00018	P.AAC:	.LONG	17694729
															00000000'	0001C		.ADDRESS	P.AAD
00	00	00	43	49	55	5F	52	45	4E	57	4F	00020	P.AAF:	.ASCII	\OWNER UIC\<0><0><0>				
															010E0009	0002C	P.AAE:	.LONG	17694729
															00000000'	00030		.ADDRESS	P.AAF
															0304 0004	00034	P.AAG:	.WORD	4, 772
															00000000G	00038		.ADDRESS	UIC_VALUE
															00000000	0003C		.LONG	0, 0
00	00	54	49	4D	49	4C	5F	4E	4F	49	53	52	45	56	00044	P.AAI:	.ASCII	\VERSION_LIMIT\<0><0><0>	
															00	00053			
															010E000D	00054	P.AAH:	.LONG	17694733
															00000000'	00058		.ADDRESS	P.AAI
00	00	54	49	4D	49	4C	5F	4E	4F	49	53	52	45	56	0005C	P.AAK:	.ASCII	\VERSION_LIMIT\<0><0><0>	
															00	0006B			

010E000D 0006C P.AAJ: .LONG 17694733  
00000000 00070 .ADDRESS P.AAK

.EXTRN SYSSGETJPIW  
.PSECT \$CODE\$,NOWRT,2

00FC 00000 GET\_QJALS:

					.WORD	Save R2,R3,R4,R5,R6,R7		0272
	57	00000000G	00	9E	00002	MOVAB	LIB\$SIGNAL, R7	
	56	00000000G	00	9E	00009	MOVAB	CLISGET VALUE, R6	
	55	00000000G	00	9E	00010	MOVAB	VRSN VALUE, R5	
	54	00000000G	00	9E	00017	MOVAB	SETFILES\$FLAGS, R4	
	53	00000000G	00	9E	0001E	MOVAB	CLISPRESENT, R3	
	52	00000000'	EF	9E	00025	MOVAB	P.AAA, R2	
	5E		10	C2	0002C	SUBL2	#16, SP	
08	AE	020E0000	8F	DD	0002F	MOVL	#34471936, DESC	0286
		OC	AE	D4	00037	CLRL	DESC+4	
			52	DD	0003A	PUSHL	R2	0291
64	01		63	01	FB	CALLS	#1, CLISPRESENT	
			02	50	FO	INSV	R0, #2, #1, SETFILES\$FLAGS	
		14		A2	9F	PUSHAB	P.AAC	0296
	63			01	FB	CALLS	#1, CLISPRESENT	
	42			50	E9	BLBC	R0, 3\$	
	64			08	88	BISB2	#8, SETFILES\$FLAGS	0299
		08		AE	9F	PUSHAB	DESC	0303
		28		A2	9F	PUSHAB	P.AAE	
	66			02	FB	CALLS	#2, CLISGET_VALUE	
	20			50	E8	BLBS	R0, 2\$	
				7E	7C	CLRQ	-(SP)	0312
		08		AE	9F	PUSHAB	IOSB	
		30		A2	9F	PUSHAB	P.AAG	
				7E	7C	CLRQ	-(SP)	
				7E	D4	CLRL	-(SP)	
00000000G	00			07	FB	CALLS	#7, SYSSGETJPIW	
	06			50	E9	BLBC	STATUS, 1\$	0313
	50			6E	3C	MOVZWL	IOSB, STATUS	0314
	17			50	E8	BLBS	STATUS, 3\$	0315
				50	DD	PUSHL	STATUS	0318
				67	11	BRB	7\$	
		00000000G	00	9F	0007C	PUSHAB	UIC VALUE	0327
		OC	AE	9F	00082	PUSHAB	DESC	
00000000G	00			02	FB	CALLS	#2, PARSE UIC	
	30			50	E9	BLBC	STATUS, 4\$	
		50		A2	9F	PUSHAB	P.AAH	0339
	63			01	FB	CALLS	#1, CLISPRESENT	
	50			50	E9	BLBC	R0, 8\$	
	64			10	88	BISB2	#16, SETFILES\$FLAGS	0342
	65	7FFF		8F	3C	MOVZWL	#32767, VRSN_VALUE	0343
		08		AE	9F	PUSHAB	DESC	0344
		68		A2	9F	PUSHAB	P.AAJ	
	66			02	FB	CALLS	#2, CLISGET_VALUE	
	3C			50	E9	BLBC	R0, 8\$	
				55	DD	PUSHL	R5	0347
		10		AE	DD	PUSHL	DESC+4	0348
		10		AE	3C	MOVZWL	DESC, -(SP)	0347
00000000G	00			03	FB	CALLS	#3, LIB\$CVT_DTB	

SETDIR  
V04-000

1 1  
16-Sep-1984 00:59:34 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:09:06 [CLIUTL.SRC]SETDIR.B32;1

Page 14  
(6)

10		50	E8	000BC		BLBS	R0, 5\$		
	08	AE	9F	000BF	4\$:	PUSHAB	DESC		0352
		01	DD	00CC2		PUSHL	#1		
	007710FA	8F	DD	000C4		PUSHL	#7803130		
67		03	FB	000CA		CALLS	#3, LIBSSIGNAL		
		1D	11	000CD		BRB	9\$		0353
50		65	D0	000CF	5\$:	MOVL	VRSN_VALUE, R0		0355
		09	19	000D2		BLSS	6\$		
0000FFFF		50	D1	000D4		CMPL	R0, #65535		0356
		0B	15	000DB		BLEQ	8\$		
	007711EA	8F	DD	000DD	6\$:	PUSHL	#7803370		0357
67		01	FB	000E3	7\$:	CALLS	#1 LIBSSIGNAL		
		04	11	000E6		BRB	0\$		
50		01	D0	000E8	8\$:	MOVL	#1, R0		0361
			04	000EB		RET			
		50	D4	000EC	9\$:	CLRL	R0		0362
			04	000EE		RET			

; Routine Size: 239 bytes, Routine Base: \$CODE\$ + 0073



```

369 0363 1 ROUTINE set_attributes (fab) =
370 0364 1 |++
371 0365 1 |
372 0366 1 | This is the routine that accesses the directory and sets the
373 0367 1 | specified attributes. If an error occurs while attempting to set
374 0368 1 | the attributes, a message telling the user is issued, and any other
375 0369 1 | directories are processed.
376 0370 1 |
377 0371 1 |--
378 0372 2 BEGIN
379 0373 2
380 0374 2 MAP
381 0375 2     fab : REF $BBLOCK;           ! Define the fab
382 0376 2
383 0377 2 LOCAL
384 0378 2     atr : BLOCKVECTOR[3,8,BYTE],  ! Attribute control block
385 0379 2     ptr,                          ! Pointer to attribute block
386 0380 2     status,                       ! Status return
387 0381 2     channel : WORD,              ! Channel number
388 0382 2     desc : $BBLOCK[dsc$c_s_bln], ! General descriptor
389 0383 2     dir_desc : $BBLOCK[dsc$c_s_bln], ! Descriptor for directory
390 0384 2     string : VECTOR[nam$c_maxrss,BYTE], ! String to build directory
391 0385 2     fib : $BBLOCK[fib$c_extdata],  ! A FIB for the QIO
392 0386 2     header : $BBLOCK[512],        ! The file header
393 0387 2     iosb : VECTOR[4,WORD];        ! I/O status block
394 0388 2
395 0389 2 BIND
396 0390 2     recattr = header[fh2$w_recattr] : $BBLOCK[atr$s_recattr],
397 0391 2
398 0392 2     nam = .fab[fab$l_nam] : $BBLOCK;   ! Define the name block
399 0393 2
400 0394 2 IF .setfile$flags[qual_quit]
401 0395 2 THEN RETURN true;
402 0396 2
403 0397 2 |
404 0398 2 | Build a descriptor to point to the directory.
405 0399 2 |
406 0400 2 | CHSMOVE(.nam[nam$b_rsl],          ! Move this many characters
407 0401 2 |     .nam[nam$l_rsa],              ! From the nam
408 0402 2 |     string);                     ! To the temp string
409 0403 2 | dir_desc[dsc$a_pointer] = string; ! Point to the string
410 0404 2 | dir_desc[dsc$w_length] = .nam[nam$b_rsl]; ! Fill in the size
411 0405 2 |
412 0406 2 | Move the final bracket over, to get the name of the final directory
413 0407 2 |
414 0408 2 | string[.nam[nam$b_dev]
415 0409 2 |     + .nam[nam$b_dir]
416 0410 2 |     + .nam[nam$b_name]] = .string[.nam[nam$b_dev]
417 0411 2 |     + .nam[nam$b_dir]
418 0412 2 |     - 1];
419 0413 2 |
420 0414 2 | string[.nam[nam$b_dev]
421 0415 2 |     + .nam[nam$b_dir] - 1] = '.'; ! Replace bracket with dot
422 0416 2 | dir_desc[dsc$w_length] = .nam[nam$b_dev]
423 0417 2 |     + .nam[nam$b_dir]
424 0418 2 |     + .nam[nam$b_name] + 1; ! Update string length
425 0419 2 |
2 | If the directory looks like [000000.something], get rid of the leading

```

```

426 0420 2 ! zeroes and dot.
427 0421 2
428 0422 2 IF NOT (HSFAIL(status = HSFIND_SUB(.dir_desc[dsc$w_length],
429 0423 2     .dir_desc[dsc$a_pointer],
430 0424 2     6,
431 0425 2     UPLIT('000000'))))
432 0426 2
433 0427 2 THEN
434 0428 2 BEGIN
435 0429 2     CHSMOVE(.dir_desc[dsc$w_length] - (.status + 7 - .dir_desc[dsc$a_pointer]),
436 0430 2     .status + 7,
437 0431 2     status);
438 0432 2     dir_desc[dsc$w_length] = .dir_desc[dsc$w_length] - 7;
439 0433 2 END;
440 0434 2
441 0435 2
442 0436 2 ! Check to see if this directory file matches the requested qualifiers
443 0437 2
444 0438 2 status = lib$qual_file_match(context, ! Call the common qualifier routine
445 0439 2     .fab, ! fab for directory
446 0440 2     0, !
447 0441 2     %ASCID 'Modify !AS? [N]: ',
448 0442 2     %REF(dir_desc));
449 0443 2
450 0444 2 IF NOT .status ! If the status is false, check
451 0445 2 THEN ! what kind of "error"
452 0446 2 BEGIN
453 0447 2     IF .status EQL lib$_filfaimat ! If no match, return
454 0448 2     THEN RETURN true
455 0449 2     ELSE IF .status EQL lib$_quipro ! If quit-right now,
456 0450 2     THEN
457 0451 2         BEGIN
458 0452 2         setfile$flags[qual_quit] = 1; ! set a flag
459 0453 2         RETURN true; ! and leave
460 0454 2         END
461 0455 2     ELSE ! For any other error,
462 0456 2     BEGIN
463 0457 2     SIGNAL(.status); ! signal it and
464 0458 2     RETURN true; ! go away
465 0459 2     END;
466 0460 2 END;
467 0461 2
468 0462 2 ! Assign a channel to the file's device
469 0463 2
470 0464 2
471 0465 2 desc[dsc$w_length] = .nam[nam$b_dev]; ! Set up the descriptor
472 0466 2 desc[dsc$a_pointer] = .nam[nam$_dev]; ! to point to the device name
473 0467 2
474 P 0468 2 IF NOT (status = $ASSIGN(
475 P 0469 2     DEVNAM = desc,
476 0470 2     CHAN = channe())
477 0471 2 THEN
478 0472 2 BEGIN
479 0473 2     file_error( set$openin,
480 0474 2     .status, .fab); ! Tell user why the assign failed
481 0475 2     RETURN true; ! And continue with other files
482 0476 2 END;

```

```
483 0477      :
484 0478      :
485 0479      : Access the directory, reading its header
486 0480      :
487 0481 desc[dsc$w_length] = fib$e_extdata;    ! Re-use descriptor to point to FIB
488 0482 desc[dsc$a_pointer] = fib;
489 0483      :
490 0484 CH$FILL(0,fib$e_extdata,fib);        ! Zero out the FIB
491 0485      :
492 0486 fib[fib$l_acctl] : fib$m_write OR        ! Set up the FIB
493 0487          fib$m_noread OR
494 0488          fib$m_nowrite;
495 0489      :
496 0490 fib[fib$w_fid_num] = .nam[nam$w_fid_num];    ! Put in the file id
497 0491 fib[fib$w_fid_seq] = .nam[nam$w_fid_seq];
498 0492 fib[fib$w_fid_rvn] = .nam[nam$w_fid_rvn];
499 0493      :
500 0494 atr[0,atr$w_type] = atr$e_header;        ! Get the file header
501 0495 atr[0,atr$w_size] = atr$e_header;
502 0496 atr[0,atr$l_addr] = header;
503 0497 atr[1,0,0,32,0] = 0;
504 0498      :
505 P 0499 status = $QIOW( CHAN = .channel,      ! Access the file, reading
506 PP 0500          FUNC = IOS_ACCESS OR IOSM_ACCESS, ! the file header
507 PP 0501          IOSB = iosb,
508      0502          P1 = desc,
509      0503          P5 = atr);
510 0504 IF .status THEN status = .iosb[0];
511 0505 IF NOT .status
512 0506 THEN file_error(set$_readerr,.status,.fab) ! If there was an error,
513 0507 ELSE                                          ! tell the user what it was
514 0508 BEGIN
515 0509 :
516 0510 ! Check to see that this is a directory file. This check is only good for
517 0511 ! an ODS2 volume.
518 0512 :
519 0513 IF .header[fh2$b_structlev] EQL 2
520 0514 AND NOT .header[fh2$v_directory]
521 0515 THEN
522 0516 BEGIN
523 0517 desc[dsc$w_length] = .nam[nam$b_rsl];
524 0518 desc[dsc$a_pointer] = .nam[nam$b_rsa];
525 0519 SIGNAL (set$_writeerr,                      ! Error writing to
526 0520          1,
527 0521          desc,                               ! This file
528 0522          set$_notdir);                       ! For this reason
529 0523 END
530 0524 ELSE
531 0525 BEGIN
532 0526 :
533 0527 ! See what qualifiers were set. Build the attribute control block
534 0528 ! as we go.
535 0529 :
536 0530 ptr = 0;                                     ! Nothing on control block
537 0531 :
538 0532 IF .setfile$flags[qual_vrsn]
539 0533 THEN
```

```

540 0534 S BEGIN
541 0535 S
542 0536 S
543 0537 S Change the version limit for the directory
544 0538 S
545 0539 S atr[0,atr$w_type] = atr$c_recattr;
546 0540 S atr[0,atr$w_size] = atr$s_recattr;
547 0541 S atr[0,atr$l_addr] = recattr;
548 0542 S recattr[fat$w_versions] = .vrsn_value;
549 0543 S ptr = .ptr + 1; ! Update the pointer
550 0544 S END;
551 0545 S
552 0546 S If .setfile$flags[qual_owner]
553 0547 S THEN
554 0548 S BEGIN
555 0549 S
556 0550 S Change the owner uic
557 0551 S
558 0552 S atr[.ptr,atr$w_type] = atr$c_uic;
559 0553 S atr[.ptr,atr$w_size] = atr$s_uic;
560 0554 S atr[.ptr,atr$l_addr] = uic_value;
561 0555 S ptr = .ptr + 1; ! Update the pointer
562 0556 S END;
563 0557 S
564 0558 S
565 0559 S Look to see if the pointer has been changed. If yes, modify the header.
566 0560 S
567 0561 S IF .ptr NEQ 0
568 0562 S THEN
569 0563 S BEGIN
570 0564 S atr[.ptr,0,0,32,0] = 0; ! Put a zero at end of list
571 0565 S status = $QIOW( CHAN = .channel, ! Make the modifications
572 0566 S FUNC = IOS MODIFY,
573 0567 S IOSB = iosb,
574 0568 S P1 = desc,
575 0569 S P5 = atr);
576 0570 S IF .status THEN status = .iosb[0];
577 0571 S IF NOT .status ! If the modify failed, tell user
578 0572 S THEN file_error(set$writeerr,.status,.fab)
579 0573 S ELSE
580 0574 S IF .setfile$flags[qual_log] ! If /LOG, tell user
581 0575 S THEN SIGNAL(set$modified,1,dir_desc);
582 0576 S $QIOW(CHAN = .channel, ! Deaccess the file
583 0577 S FUNC = IOS DEACCESS,
584 0578 S IOSB = iosb,
585 0579 S P1 = desc);
586 0580 S END; ! End of header modify
587 0581 S END; ! End of qualifier scan
588 0582 S
589 0583 S Deassign the channel
590 0584 S
591 0585 S IF NOT (status = $DASSGN(CHAN = .channel))
592 0586 S THEN file_error(set$closeerr, .status, .fab);
593 0587 S END;
594 0588 S
595 0589 S RETURN true;
596 0590 S END;

```

															.PSECT \$PLITS,NOWRT,NOEXE,2				
5D	4E	5B	20	3F	53	41	00	00	30	30	30	30	30	00074	P.AAL:	.ASCII	\000000\<0><0>	:	
							21	20	79	66	69	64	6F	4D	0007C	P.AAN:	.ASCII	\Modify !AS? [N]: \<0><0><0>	:
										00	00	00	20	3A	0008B				:
													010E0011	00090	P.AAM:	.LONG	17694737	:	
												00000000'	00094			.ADDRESS	P.AAN	:	
															.EXTRN SYSS\$ASSIGN, SYSS\$QIOW				
															.EXTRN SYSS\$DASSGN				
															.PSECT \$CODES,NOWRT,2				
															OFFC 00000 SET_ATTRIBUTES:				
										5B	00000000G	00	9E	00002		.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0363	
										5A	00000000G	00	9E	00009		MOVAB	SYSS\$QIOW, R11	:	
										59	00000000G	00	9E	00010		MOVAB	LIB\$SIGNAL, R10	:	
										5E	FCA8	CE	9E	00017		MOVAB	SETFILES\$FLAGS, R9	:	
										58	04	AC	00	0001C		MOVL	-856(SP), SP	:	
										56	28	A9	00	00020		MOVL	FAB, R8	: 0392	
										69		01	E1	00024		MOVL	40(R8), R6	:	
												023E	31	00028		BBC	#1 SETFILES\$FLAGS, 1\$	: 0394	
																BRW	19\$	:	
																MOVZBL	3(R6), R0	: 0400	
																MOVZBL	R0, @4(R6), STRING	:	
																MOVAB	STRING, DIR_DESC+4	: 0403	
																MOVZBW	3(R6), DIR_DESC	: 0404	
																MOVZBL	57(R6), R0	: 0409	
																MOVZBL	58(R6), R1	:	
																ADDL2	R1, R0	:	
																MOVZBL	59(R6), R1	: 0410	
																ADDL2	R0, R1	:	
																MOVAB	STRING-1[R0], STRING[R1]	:	
																MOVAB	#46, STRING-1[R0]	: 0414	
																ADDW3	#1, R1, DIR_DESC	: 0417	
																MATCHC	#6, P.AAL, DIR_DESC, @DIR_DESC+4	: 0425	
																BEQL	2\$	:	
																MOVL	#6, R3	:	
																SUBL2	#2, R3	: 2\$:	
																MOVAL	-(R3), STATUS	:	
																BE2L	3\$	:	
																SURL3	STATUS, DIR_DESC+4, R0	: 0429	
																MOVZWL	DIR_DESC, RT	:	
																ADDL2	R1, R0	:	
																SUBL2	#7, R0	:	
																MOVL	STATUS, R3	: 0430	
																MOVZBL	R0, 7(R3), (STATUS)	: 0431	
																SUBW2	#7, DIR_DESC	: 0432	
																MOVAB	DIR_DESC, (SP)	: 0442	
																PUSHL	SP	:	
																PUSHAB	P.AAM	: 0440	
																CLRL	-(SP)	: 0438	
																PUSHL	R8	: 0439	
																PUSHAB	CONTEXT	: 0438	

00000000G	00		05	FB	000B0	CALLS	#5, LIB\$QUAL_FILE_MATCH			
	57		50	DO	000B7	MOVL	R0, STATUS			
	1F		57	EB	000BA	BLBS	STATUS, 6\$		0444	
00000000G	8F		57	D1	000BD	CMPL	STATUS, #LIBS_FILFAIMAT		0447	
			13	13	000C4	BEQL	5\$			
00000000G	8F		57	D1	000C6	CMPL	STATUS, #LIBS_QUIPRO		0449	
			05	12	000CD	BNEQ	4\$			
	69		02	88	000CF	BISB2	#2, SETFILES\$FLAGS		0452	
			05	11	000D2	BRB	5\$		0453	
			57	DD	000D4	4\$: PUSHL	STATUS		0457	
	6A		01	FB	000D6	CALLS	#1, LIB\$SIGNAL			
			01	8D	31	000D9	5\$: BRW	19\$	0458	
	E0	AD	39	A6	9B	000DC	6\$: MOVZBW	57(R6), DESC	0465	
	E4	AD	44	A6	DO	000E1	MOVL	68(R6), DESC+4	0466	
				7E	7C	000E6	CLRQ	-(SP)	0470	
			0C	AE	9F	000E8	PUSHAB	CHANNEL		
			E0	AD	9F	000EB	PUSHAB	DESC		
00000000G	00		04	FB	000EE	CALLS	#4, SYSS\$ASSIGN			
	57		50	DO	000F5	MOVL	R0, STATUS			
	0B		57	EB	000F8	BLBS	STATUS, 7\$			
	7E		57	7D	000FB	MOVQ	STATUS, -(SP)		0474	
		0077109A	8F	DD	000FE	PUSHL	#7803034		0473	
			69	11	00104	BRB	9\$			
	E0	AD	20	B0	00106	7\$: MOVW	#32, DESC		0481	
	E4	AD	FEB8	CD	9E	0010A	MOVAB	FIB, DESC+4	0482	
			6E	00	2C	00110	MOVCS	#0, (SP), #0, #32, FIB	0484	
				CD	00115					
	FEB8	CD	0501	8F	3C	00118	MOVZWL	#1281, FIB	0487	
	FEB8	CD	24	A6	DO	0011F	MOVL	36(R6), FIB+4	0490	
	FECO	CD	28	A6	B0	00125	MOVW	40(R6), FIB+8	0492	
	E8	AD	000A02C0	8F	DO	0012B	MOVL	#655872, ATR	0495	
	EC	AD	10	AE	9E	00133	MOVAB	HEADER, ATR+4	0496	
			F0	AD	D4	00138	CLRL	ATR+8	0497	
				7E	D4	0013B	CLRL	-(SP)	0503	
			E8	AD	9F	0013D	PUSHAB	ATR		
				7E	7C	00140	CLRQ	-(SP)		
				7E	D4	00142	CLRL	-(SP)		
			E0	AD	9F	00144	PUSHAB	DESC		
				7E	7C	00147	CLRQ	-(SP)		
			28	AE	9F	00149	PUSHAB	IOSB		
			7E	72	8F	9A	0014C	MOVZBL	#114, -(SP)	
			7E	2C	AE	3C	00150	MOVZWL	CHANNEL, -(SP)	
				7E	D4	00154	CLRL	-(SP)		
	6B		0C	FB	00156	CALLS	#12, SYSS\$QIOW			
	57		50	DO	00159	MOVL	R0, STATUS			
	07		57	E9	0015C	BLBC	STATUS, 8\$		0504	
	57		08	AE	3C	0015F	MOVZWL	IOSB, STATUS		
	0C		57	EB	00163	BLBS	STATUS, 10\$		0505	
	7E		57	7D	00166	8\$: MOVQ	STATUS, -(SP)		0506	
		00000000G	8F	DD	00169	PUSHL	#SETS_READERR			
			00F0	31	0016F	9\$: BRW	18\$			
			02	17	AE	91	00172	10\$: CMPB	HEADER+7, #2	0513
				26	12	00176	BNEQ	11\$		
				05	E0	00178	BBS	#5, HEADER+53, 11\$	0514	
21	45	AE		A6	9B	0017D	MOVZBW	3(R6), DESC	0517	
	E0	AD	03	A6	DO	00182	MOVL	4(R6), DESC+4	0518	
	E4	AD	04	A6	DO	00182	MOVL	4(R6), DESC+4	0518	
			00000000G	8F	DD	00187	PUSHL	#SETS_NOTDIR	0519	

		E0	AD	9F	0018D	PUSHAB	DESC		
			01	DD	00190	PUSHL	#1		
		00000000G	8F	DD	00192	PUSHL	#SETS_WRITEERR		
	6A		04	FB	00198	CALLS	#4, LIBSSIGNAL		
			00AA	31	0019B	BRW	17\$		0513
			50	D4	0019E	CLRL	PTR		0530
17		69	04	E1	001A0	BBC	#4, SETFILESFLAGS, 12\$		0532
	EB	AD	00040020	8F	DO	001A4	MOVL	#262176, ATR	0540
	EC	AD	24	AE	9E	001AC	MOVAB	RECATR, ATR+4	0541
	42	AE	00000000G	00	B0	001B1	MOVW	VRSN_VALUE, RECATR+30	0542
				50	D6	001B9	INCL	PTR	0543
1B		69		03	E1	001BB	BBC	#3, SETFILESFLAGS, 13\$	0546
			EA	AD40	7F	001BF	PUSHAQ	ATR+2[PTR]	0552
		9E		15	B0	001C3	MOVW	#21, @(SP)+	
			E8	AD40	7F	001C6	PUSHAQ	ATR[PTR]	0553
		9E		04	B0	001CA	MOVW	#4, @(SP)+	
			EC	AD40	7F	001CD	PUSHAQ	ATR+4[PTR]	0554
		9E	00000000G	00	9E	001D1	MOVAB	UIC_VALUE, @(SP)+	
				50	D6	001D8	INCL	PTR	0555
				6C	13	001DA	BEQL	17\$	0561
			E8	AD40	7F	001DC	PUSHAQ	ATR[PTR]	0564
				9E	D4	001E0	CLRL	@(SP)+	
				7E	D4	001E2	CLRL	-(SP)	0569
			E8	AD	9F	001E4	PUSHAB	ATR	
				7E	7C	001E7	CLRQ	-(SP)	
				7E	D4	001E9	CLRL	-(SP)	
			E0	AD	9F	001EB	PUSHAB	DESC	
				7E	7C	001EE	CLRQ	-(SP)	
			28	AE	9F	001F0	PUSHAB	IOSB	
				36	DD	001F3	PUSHL	#54	
		7E	2C	AE	3C	001F5	MOVZWL	CHANNEL, -(SP)	
				7E	D4	001F9	CLRL	-(SP)	
		6B		0C	FB	001FB	CALLS	#12, SYSSQIOW	
		57		50	DO	001FE	MOVL	RO, STATUS	
		07		57	E9	00201	BLBC	STATUS, 14\$	0570
		57	08	AE	3C	00204	MOVZWL	IOSB, STATUS	
		12		57	E8	00208	BLBS	STATUS, 15\$	0571
		7E		57	7D	0020B	MOVQ	STATUS, -(SP)	0572
			00000000G	00	8F	DD	0020E	PUSHL	#SETS_WRITEERR
					03	FB	00214	CALLS	#3, FILE_ERROR
					12	11	0021B	BRB	16\$
0E		69		02	E1	0021D	BBC	#2, SETFILESFLAGS, 16\$	0574
			D8	AD	9F	00221	PUSHAB	DIR_DESC	0575
				01	DD	00224	PUSHL	#1	
			00000000G	8F	DD	00226	PUSHL	#SETS_MODIFIED	
		6A		03	FB	0022C	CALLS	#3, LIBSSIGNAL	
				7E	7C	0022F	CLRQ	-(SP)	0579
				7E	7C	00231	CLRQ	-(SP)	
				7E	D4	00233	CLRL	-(SP)	
			E0	AD	9F	00235	PUSHAB	DESC	
				7E	7C	00238	CLRQ	-(SP)	
			28	AE	9F	0023A	PUSHAB	IOSB	
				34	DD	0023D	PUSHL	#52	
		7E	2C	AE	3C	0023F	MOVZWL	CHANNEL, -(SP)	
				7E	D4	00243	CLRL	-(SP)	
		6B		0C	FB	00245	CALLS	#12, SYSSQIOW	
		7E	04	AE	3C	00248	MOVZWL	CHANNEL, -(SP)	0585

SETDIR  
V04-000

D 2  
16-Sep-1984 00:59:34 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:09:06 [CLIUTL.SRC]SETDIR.B32;1

Page 22  
(7)

00000000G	00	01	FB	0024C	CALLS	#1, SYSSDASSGN	:
	57	50	DO	00253	MOVL	R0, STATUS	:
	10	57	EB	00256	B_BS	STATUS, 19\$	:
	7E	57	7D	00259	P^VQ	STATUS, -(SP)	0586
		8F	DD	0025C	PUSHL	#SET\$ CLOSEERR	:
00000000G	00	03	FB	00262	CALLS	#3, FILE_ERROR	:
	50	01	DO	00269	MOVL	#1, R0	0589
		04	0026C	RET			0590

; Routine Size: 621 bytes, Routine Base: \$CODE\$ + 0162



```

: 598 0591 1 ROUTINE get_dir (dir_desc) =
: 599 0592 1 +-
: 600 0593 1 |
: 601 0594 1 | This routine is called to get each directory spec from the command line.
: 602 0595 1 | When there are no more directories, a value of FALSE is returned.
: 603 0596 1 |
: 604 0597 1 | --
: 605 0598 2 BEGIN
: 606 0599 2
: 607 0600 2 MAP dir_desc : REF $BBLOCK;
: 608 0601 2
: 609 0602 2 LOCAL
: 610 0603 2 desc : $BBLOCK[dsc$_s_bln], ! General descriptor
: 611 0604 2 endchar : BYTE, ! End-of-directory character
: 612 0605 2 eos, ! End of string
: 613 0606 2 ptr, ! Pointer in string
: 614 0607 2 str_ptr, ! Pointer to rest of string
: 615 0608 2 str_len, ! Length of rest of string
: 616 0609 2 temp_string : VECTOR[nam$_maxrss, BYTE], ! The temporary string
: 617 0610 2 temp,
: 618 0611 2 status;
: 619 0612 2
: 620 0613 2 IF .setfile$flags[qual_quit]
: 621 0614 2 THEN RETURN false;
: 622 0615 2
: 623 0616 2 |
: 624 0617 2 | Check here to see if a trailing ellipse is being treated. If so,
: 625 0618 2 | then FILE_FAB[FAB$_CTX] will be set to TRUE, and there's no need
: 626 0619 2 | to search and see if such a trailing ellipse is present. However,
: 627 0620 2 | if the value is set to FALSE, then get a new directory spec.
: 628 0621 2 |
: 629 0622 2 IF NOT .file_fab[fab$_ctx] ! If not processing an ellipse
: 630 0623 2 THEN ! then get the next directory
: 631 0624 2 BEGIN
: 632 0625 2 IF NOT cli$get_value(%ASCII 'DIRECTORY', .dir_desc)
: 633 0626 2 THEN RETURN false;
: 634 0627 2
: 635 0628 2 file_fab[fab$_fna] = .dir_desc[dsc$_a_pointer];
: 636 0629 2 file_fab[fab$_fns] = .dir_desc[dsc$_w_length];
: 637 0630 2
: 638 0631 2 |
: 639 0632 2 | Since this is a new entry, it must be checked for a trailing ellipse.
: 640 0633 2 |
: 641 0634 2 | CHSMOVE( .dir_desc[dsc$_w_length], ! Move this many chars
: 642 0635 2 | .dir_desc[dsc$_a_pointer], ! From the CLI area
: 643 0636 2 | temp_string); ! To the temp string
: 644 0637 2 | str_ptr = temp_string; ! Set up pointer
: 645 0638 2 | str_len = .dir_desc[dsc$_w_length]; ! and length.
: 646 0639 2 |
: 647 0640 2 | Look for ellipses.
: 648 0641 2 |
: 649 0642 2 | WHILE NOT CH$FAIL(temp = CH$FIND_SUB(.str_len, .str_ptr, 3, UPLIT('...')))
: 650 0643 2 | DO
: 651 0644 2 | BEGIN
: 652 0645 2 | str_ptr = .temp + 3; ! Update pointer
: 653 0646 2 | str_len = .str_len - (.temp - .str_ptr) - 3;
: 654 0647 2 | END;

```

```

655 0648
656 0649
657 0650
658 0651
659 0652
660 0653
661 0654
662 0655
663 0656
664 0657
665 0658
666 0659
667 0660
668 0661
669 0662
670 0663
671 0664
672 0665
673 0666
674 0667
675 0668
676 0669
677 0670
678 0671
679 0672
680 0673
681 0674
682 0675
683 0676
684 0677
685 0678
686 0679
687 0680
688 0681
689 0682
690 0683
691 0684
692 0685
693 0686
694 0687
695 0688
696 0689
697 0690
698 0691
699 0692
700 0693
701 0694
702 0695
703 0696
704 0697
705 0698
706 0699
707 0700
708 0701
709 0702
710 0703
711 0704

:
: After the final ellipse, check to see if it is at the end of the
: directory specification. If so, then change the context field of
: the fab, and insert an end bracket at the beginning of the ellipse.
:
: IF (.str_ptr EQL temp_string + .file_fab[fab$b_fns] - 1)
: THEN
: BEGIN
:   file_fab[fab$l_ctx] = true;          ! Show that there's a trailing ellipse
:   CH$WCHAR(.str_ptr, .str_ptr - 3);  ! Put the end bracket in place
:   file_fab[fab$l_fna] = temp_string; ! Set up FAB fields
:   file_fab[fab$b_fns] = .str_ptr - 3 - temp_string + 1;
: END;
:
: END
:
: If here, then the trailing ellipse has been processed, and this is the
: second time thru. Restore the original file name.
:
: ELSE
: BEGIN
:   file_fab[fab$l_fna] = .dir_desc[dsc$a_pointer]; ! Original filename
:   file_fab[fab$b_fns] = .dir_desc[dsc$w_length]; ! Original length
:   file_fab[fab$l_ctx] = false;                 ! Ellipse processed
: END;
:
: Parse the input string
:
: $NAM_INIT (NAM = file_rlf);          ! Re-init the RLF
: IF (.file_nam[nam$b_dev] NEQ 0)      ! If a device was
: THEN                                  ! specified, then
: BEGIN
:   file_fab[fab$l_dna] = .file_nam[nam$l_dev]; ! Make device sticky
:   file_fab[fab$b_dns] = .file_nam[nam$b_dev];
: END;
: IF NOT (status = $PARSE(FAB = file_fab))
: THEN
: BEGIN
:   desc[dsc$w_length] = .file_fab[fab$b_fns];
:   desc[dsc$a_pointer] = .file_fab[fab$l_fna];
:   SIGNAL_STOP(set$syntax,
:               1, desc, .status);
: END;
:
: Check the parsed string for legality, i.e. nothing after the directory
:
: IF (.file_nam[nam$b_name] NEQ 0 OR
:     .file_nam[nam$b_type] NEQ 1 OR
:     .file_nam[nam$b_ver] NEQ 1 )
: THEN
: BEGIN
:   desc[dsc$w_length] = .file_fab[fab$b_fns];
:   desc[dsc$a_pointer] = .file_fab[fab$l_fna];
:   SIGNAL_STOP(set$syntax,
:               1, desc, ss$_badirectory);
: END;

```

```
712 0705 2 :  
713 0706 2 : Check if a search list is used. If so SIGNAL, as search lists are not  
714 0707 2 : allowed (for devices) in this application.  
715 0708 2 :  
716 0709 2 IF .file_nam[nam$u_search_list]  
717 0710 2 THEN  
718 0711 2     SIGNAL_STOP(set$_filspcrch, 1, .dir_desc, set$_nosrchlst);  
719 0712 2 :  
720 0713 2 : Determine what the directory terminator character was, and save it.  
721 0714 2 :  
722 0715 2 endchar = .(.file_nam[nam$l_dir] + .file_nam[nam$b_dir] - 1);  
723 0716 2 :  
724 0717 2 :  
725 0718 2 : The directory string must now be analyzed and manipulated so that the  
726 0719 2 : final directory entry becomes a file. First, initialize some pointers.  
727 0720 2 :  
728 0721 2 desc[dsc$w_length] = .file_nam[nam$b_esl] - 2;  
729 0722 2 desc[dsc$a_pointer] = .file_nam[nam$l_esa];  
730 0723 2 str_ptr = .desc[dsc$a_pointer];  
731 0724 2 str_len = .desc[dsc$w_length];  
732 0725 2 ptr = 0;  
733 0726 2 eos = .desc[dsc$a_pointer] + .desc[dsc$w_length] - 1;  
734 0727 2 :  
735 0728 2 :  
736 0729 2 : Look for wildcard ellipses  
737 0730 2 :  
738 0731 2 WHILE NOT CH$FAIL(temp = CH$FIND_SUB(.str_len, .str_ptr, 3, uplit('...')))  
739 0732 2 DO  
740 0733 2     BEGIN  
741 0734 2 :  
742 0735 2 : Make PTR point to the beginning of the "...", and advance the string  
743 0736 2 : pointer to the character just past the "..."  
744 0737 2 :  
745 0738 2     ptr = .temp;  
746 0739 2     str_len = .str_len - (.temp - .str_ptr) - 3;  
747 0740 2     str_ptr = .temp + 3;  
748 0741 2     END;  
749 0742 2 :  
750 0743 2 :  
751 0744 2 : If there was any occurrence of "...", point just past it.  
752 0745 2 :  
753 0746 2 IF .ptr NEQ 0 THEN ptr = .ptr + 3;  
754 0747 2 :  
755 0748 2 :  
756 0749 2 : Find the last directory in the specification  
757 0750 2 :  
758 0751 2 WHILE NOT CH$FAIL(temp = CH$FIND_CH(.str_len, .str_ptr, '.'))  
759 0752 2 DO  
760 0753 2     BEGIN  
761 0754 2 :  
762 0755 2 : Make PTR point to the ".", and advance the string pointer to  
763 0756 2 : the first character after the "."  
764 0757 2 :  
765 0758 2     ptr = .temp;  
766 0759 2     str_len = .str_len - (.temp - .str_ptr) - 1;  
767 0760 2     str_ptr = .temp + 1;  
768 0761 2     END;
```

```

769 0762 2
770 0763 2 IF .ptr NEQ 0
771 0764 2 THEN
772 0765 2 BEGIN
773 0766 2
774 0767 2 If here, then either a trailing ellipse, or a final sub-directory
775 0768 2 was specified. If the pointer is at the bracket, then there is a
776 0769 2 trailing ellipse, in which case only a '*' is required.
777 0770 2
778 0771 2 IF .ptr EQL .eos
779 0772 2 THEN
780 0773 2 BEGIN
781 0774 2 CHSA WCHAR('*', ptr); ! Stick an asterisk after the bracket.
782 0775 2 ptr = .ptr + 1; ! Adjust the pointer.
783 0776 2 END
784 0777 2
785 0778 2 If the pointer is inside the bracket, then the last directory name
786 0779 2 must be moved out of the brackets.
787 0780 2
788 0781 2 ELSE
789 0782 2 BEGIN
790 0783 2
791 0784 2 Check to see if the directory is [main.sub] or [main...sub]
792 0785 2
793 0786 2 IF .ptr EQLU .str_ptr
794 0787 2 THEN
795 0788 2 BEGIN ! [main...sub] form
796 0789 2 str len = .eos - .ptr;
797 0790 2 CH$MOVE(.str_len, .ptr, .ptr+1);
798 0791 2 CH$WCHAR(.endchar, .ptr);
799 0792 2 ptr = .ptr + .str_len + 1;
800 0793 2 END ! end of [main...sub] processing
801 0794 2 ELSE
802 0795 2 BEGIN ! [main.sub] form
803 0796 2 str len = .eos - .str_ptr;
804 0797 2 CH$WCHAR A(.endchar, ptr);
805 0798 2 ptr = .ptr + .str_len;
806 0799 2 END; ! end of [main.sub] processing
807 0800 2 END;
808 0801 2 END ! End of non-zero pointer stuff
809 0802 2 ELSE
810 0803 2 BEGIN
811 0804 2
812 0805 2 If the pointer is still zero, then there is either a wildcard, a main
813 0806 2 directory, or a [g,m] directory. In all such cases, a main directory
814 0807 2 of [000000] must be fabricated.
815 0808 2
816 0809 2 status = CH$FIND_CH(.str_len, .str_ptr, ','); ! Save for later
817 0810 2
818 0811 2 Move the string out seven spaces and insert "000000]"
819 0812 2
820 0813 2 str_ptr = .desc[dsc$a_pointer] + .file_nam[nam$b dev] + 1;
821 0814 2 temp = CH$MOVE(.eos - .str_ptr, .str_ptr, .str_ptr + 7);
822 0815 2 str_ptr = CH$MOVE(6, UPLITT('000000'), .str_ptr);
823 0816 2 CH$MOVE(1, endchar, .str_ptr);
824 0817 2
825 0818 2 If no comma was found, then all that is required is to update the

```

```

826 0819 3 : pointer.
827 0820 3 :
828 0821 3 :   IF CH$FAIL(.status) THEN ptr = .temp
829 0822 3 :
830 0823 3 : Otherwise, it's a [g,m] directory. Convert it.
831 0824 3 :
832 0825 3 :   ELSE
833 0826 4 :     BEGIN
834 0827 4 :
835 0828 4 :     LOCAL tparse_block : $BBLOCK[tpa$k_length0];      ! Define a TPARSE block
836 0829 4 :
837 0830 4 :     CH$FILL(0, tpa$k_length0, tparse_block);          ! Zero it.
838 0831 4 :     tparse_block[tpa$l_count] = tpa$k_count0;        ! Fill in size
839 0832 4 :
840 0833 4 :     tparse_block[tpa$l_stringcnt] = .eos - .str_ptr;
841 0834 4 :     tparse_block[tpa$l_stringptr] = .str_ptr + 7;
842 0835 5 :     IF NOT (status = lib$tparse(tparse_block,
843 0836 5 :                               dir_state,
844 0837 5 :                               dir_keys))
845 0838 4 :     THEN SIGNAL_STOP(.status)
846 0839 4 :     ELSE
847 0840 5 :       BEGIN
848 0841 5 :         LOCAL temp_desc : $BBLOCK[dsc$c_s_bln];
849 0842 5 :         temp_desc[dsc$w_length] = 6;
850 0843 5 :         temp_desc[dsc$a_pointer] = .str_ptr + 7;
851 0844 6 :         IF NOT (status = SY$FAO($descriptor('!2(30W)'),
852 0845 6 :                               temp_desc,
853 0846 6 :                               temp_desc,
854 0847 6 :                               .group,
855 0848 6 :                               .member))
856 0849 5 :         THEN SIGNAL_STOP(.status)
857 0850 5 :         ELSE ptr = .str_ptr + 14;
858 0851 4 :         END;
859 0852 3 :       END;
860 0853 2 :     END;
861 0854 2 :     ptr = CH$MOVE(4, UPLIT('.DIR'), .ptr);
862 0855 2 :     file_fab[fab$b_fns] = .ptr - .desc[dsc$a_pointer];
863 0856 2 :     file_fab[fab$l_fna] = .desc[dsc$a_pointer];
864 0857 2 :
865 0858 2 : RETURN true;
866 0859 1 : END;

```

										.PSECT		\$SPLITS,NOWRT,NOEXE,2				
00	00	00	59	52	4F	54	43	45	52	49	44	00098	P.AAP:	.ASCII	\DIRECTORY\<0><0><0>	:
										010E0009		000A4	P.AAO:	.LONG	17694729	:
										00000000		000A8		.ADDRESS	P.AAP	:
								00	2E	2E	2E	000AC	P.AAQ:	.ASCII	\...\<0>	:
								00	2E	2E	2E	000B0	P.AAR:	.ASCII	\...\<0>	:
	00	00	30	30	30	30	30	30	30	30	30	000B4	P.AAS:	.ASCII	\000000\<0><0>	:
		29	57	4F	33	28	32	21				000BC	P.AAU:	.ASCII	\!2(30W)\	:
												000C3		.BLKB	1	:
										00000007		000C4	P.AAT:	.LONG	7	:
										00000000		000C8		.ADDRESS	P.AAU	:
					52	49	44	2E				000CC	P.AAV:	.ASCII	\.DIR\	:

		SRMS_PTR=	FILE_RLF		
		.EXTRN	SYSSPARSE		
		.PSECT	\$CODE\$,NOWRT,2		
		OFFC 00000	GET_DIR: .WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0591
		SE FEC4 CE 9E 00002	MOVAB	-316(SP), SP	
03	00000000G	00 01 E1 00007	BBC	#1, SETFILE\$FLAGS, 2\$	: 0613
		56 04 02F8 31 0000F	BRW	29\$	
		03 00000000G 00 AC DO 00012	MOVL	DIR_DESC, R6	: 0625
		008A 31 00016	BLBC	FILE_FAB+24, 3\$	: 0622
		56 DD 00020	BRW	7\$	
		00000000' EF 9F 00022	3\$: PUSHL	R6	: 0625
	00000000G	00 02 FB 00028	PUSHAB	P.AAO	
		DD 50 E9 0002F	CALLS	#2, CLISGET_VALUE	
	00000000G	00 04 A6 DO 00032	BLBC	R0, 1\$	
	00000000G	00 66 90 0003A	MOVL	4(R6), FILE_FAB+44	: 0628
34	AE 04	B6 66 28 00041	MOVB	(R6), FILE_FAB+52	: 0629
		5A 34 AE 9E 00047	MOVC3	(R6), @4(R6), TEMP_STRING	: 0634
		57 66 3C 0004B	MOVAB	TEMP_STRING, STR_PTR	: 0637
6A	57 00000000'	EF 03 39 0004E	MOVZWL	(R6), STR_LEN	: 0638
		53 03 13 00057	4\$: MATCHC	#3, P.AAO, STR_LEN, (STR_PTR)	: 0642
		5B 03 DO 00059	BEQL	5\$	
		53 D7 0005C	5\$: MOVL	#3, R3	
		5B 73 3E 0005E	DECL	R3	
		5A 03 AB 9E 00063	MOVAV	-(R3), TEMP	
		5A 5B C3 00067	BEQL	6\$	
53	57 FD A347	9E 0006B	MOVAB	3(R11), STR_PTR	: 0645
		DC 11 00070	SUBL3	TEMP, STR_PTR, R3	: 0646
		5C 00000000G 00 9A 00072	MOVAB	-3(R3)[STR_LEN], STR_LEN	
		50 33 AE40 9E 00079	BRB	4\$	: 0642
		50 5A D1 0007E	6\$: MOVZBL	FILE_FAB+52, R0	: 0653
		00000000G 00 01 DO 00083	MOVAB	TEMP_STRING-1[R0], R0	
		51 FD AA 9E 0008A	CML	STR_PTR, R0	
		61 6A 90 0008E	BNEQ	8\$	
		00000000G 00 34 AE 9E 00091	MOVL	#1, FILE_FAB+24	: 0656
		50 34 AE 9E 00099	MOVAB	-3(STR_PTR), R1	: 0657
		50 51 C2 0009D	MOVB	(STR_PTR), (R1)	
	00000000G 00	01 50 83 000A0	MOVAB	TEMP_STRING, FILE_FAB+44	: 0658
		00000000G 00 04 A6 DO 000AA	MOVAB	TEMP_STRING, R0	: 0659
		00000000G 00 66 90 000B2	SUBL2	R1, R0	
		00000000G 00 00 D4 000B9	SUBB3	R0, #1, FILE_FAB+52	
		00000000G 00 00 2C 000BF	BRB	8\$	: 0622
0060	8F 00	6E 00 2C 000BF	7\$: MOVL	4(R6), FILE_FAB+44	: 0668
		00000000' EF 8F 80 000CB	MOVB	(R6), FILE_FAB+52	: 0669
		00000000' EF 6002 8F 80 000CB	CLRL	FILE_FAB+24	: 0670
		50 00000000G 00 9A 000D4	8\$: MOVC5	#0, (SP), #0, #96, SRMS_PTR	: 0676
		00000000G 00 12 13 000DB	MOVW	#24578, SRMS_PTR	
		00000000G 00 00 DO 000DD	MOVZBL	FILE_NAM+57, R0	: 0677
		00000000G 00 50 90 000EB	BEQL	9\$	
		00000000G 00 00 9F 000EF	9\$: MOVL	FILE_NAM+68, FILE_FAB+48	: 0680
		00000000G 00 01 FB 000F5	MOVB	R0, FILE_FAB+53	: 0681
		6E 50 DO 000FC	PUSHAB	FILE_FAB	: 0683
			CALLS	#1, SYSSPARSE	
			MOVL	R0, STATUS	

		24	6E	E8	000FF	BLBS	STATUS, 10\$	
	F8	AD 00000000G	00	9B	00102	MOVZBW	FILE_FAB+52, DESC	0686
	FC	AD 00000000G	00	D0	0010A	MOVL	FILE_FAB+44, DESC+4	0687
			6E	DD	00112	PUSHL	STATUS	0689
		F8	AD	9F	00114	PUSHAB	DESC	0688
			01	DD	00117	PUSHL	#1	
		007710FA	8F	DD	00119	PUSHL	#7803130	
	00000000G	00	04	FB	0011F	CALLS	#4, LIB\$STOP	
		00000000G	00	95	00126	TSTB	FILE_NAM+59	0695
			12	12	0012C	BNEQ	11\$	
		01 00000000G	00	91	0012E	CMPB	FILE_NAM+60, #1	0696
			09	12	00135	BNEQ	11\$	
		01 00000000G	00	91	00137	CMPB	FILE_NAM+61, #1	0697
			27	13	0013E	BEQL	12\$	
	F8	AD 00000000G	00	9B	00140	MOVZBW	FILE_FAB+52, DESC	0700
	FC	AD 00000000G	00	D0	00148	MOVL	FILE_FAB+44, DESC+4	0701
		7E 0828	8F	3C	00150	MOVZWL	#2088, -(SP)	0702
		F8	AD	9F	00155	PUSHAB	DESC	
			01	DD	00158	PUSHL	#1	
		007710FA	8F	DD	0015A	PUSHL	#7803130	
	00000000G	00	04	FB	00160	CALLS	#4, LIB\$STOP	
17	00000000G	00	03	E1	00167	BBC	#3, FILE_NAM+53, 13\$	0709
		00771352	8F	DD	0016F	PUSHL	#7803730	0711
			56	DD	00175	PUSHL	R6	
			01	DD	00177	PUSHL	#1	
		0077134A	8F	DD	00179	PUSHL	#7803722	
	00000000G	00	04	FB	0017F	CALLS	#4, LIB\$STOP	
		50 00000000G	00	9A	00186	MOVZBL	FILE_NAM+58, R0	0715
		50 00000000G	00	C0	0018D	ADDL2	FILE_NAM+72, R0	
			04	AE	FF A0	MOVB	-1(R0), ENDCHAR	
	F8	AD 00000000G	00	9B	00199	MOVZBW	FILE_NAM+11, DESC	0721
	F8	AD	02	A2	001A1	SUBW2	#2, DESC	
	FC	AD 00000000G	00	D0	001A5	MOVL	FILE_NAM+12, DESC+4	0722
		59 FC	AD	D0	001AD	MOVL	DESC+4, R9	0723
		5A	59	D0	001B1	MOVL	R9, STR_PTR	
		50 F8	AD	3C	001B4	MOVZWL	DESC, R0	0724
		57	50	D0	001B8	MOVL	R0, STR_LEN	
			58	D4	001BB	CLRL	PTR	0725
		56 FF	A049	9E	001BD	MOVAB	-1(R0)[R9], EOS	0726
6A	57 00000000'	EF	03	39	001C2	MATCHC	#3, P.AAR, STR_LEN, (STR_PTR)	0731
			03	13	001CB	BEQL	15\$	
		53	03	D0	001CD	MOVL	#3, R3	
			53	D7	001D0	DECL	R3	15\$:
		5B	73	3E	001D2	MOVAV	-(R3), TEMP	
			12	13	001D5	BEQL	16\$	
		58	5B	D0	001D7	MOVL	TEMP, PTR	0738
53		5A	5B	C3	001DA	SUBL3	TEMP, STR_PTR, R3	0739
		57	FD A347	9E	001DE	MOVAB	-3(R3)[STR_LEN], STR_LEN	
		5A	03	AB	001E3	MOVAB	3(R11), STR_PTR	0740
			D9	11	001E7	BRB	14\$	0731
			58	D5	001E9	TSTL	PTR	0746
			03	13	001EB	BEQL	17\$	
		58	03	C0	001ED	ADDL2	#3, PTR	
6A		57	2E	3A	001F0	LOCC	#46, STR_LEN, (STR_PTR)	0751
			02	12	001F4	BNEQ	18\$	
			51	D4	001F6	CLRL	R1	
		5B	51	D0	001F8	MOVL	R1, TEMP	18\$:

			12	13	001FB	BEQL	19\$				
		58	5B	D0	001FD	MOVL	TEMP, PTR		0758		
53		5A	5B	C3	00200	SUBL3	TEMP, STR_PTR, R3		0759		
		57	FF A347	9E	00204	MOVAB	-1(R3)[STR_LEN], STR_LEN				
		5A	01	AB	9E	00209	MOVAB	1(R11), STR_PTR	0760		
				E1	11	0020D	BRB	17\$	0751		
				58	D5	0020F	TSTL	PTR	0763		
				32	13	00211	BEQL	22\$			
		56		58	D1	00213	CMPL	PTR, EOS	0771		
				07	12	00216	BNEQ	20\$			
				58	D6	00218	INCL	PTR	0774		
		88		2A	90	0021A	MOVB	#42, (PTR)+			
				5F	11	0021D	BRB	24\$	0771		
		5A		58	D1	0021F	CMPL	PTR, STR_PTR	0786		
				14	12	00222	BNEQ	21\$			
01	57	56		58	C3	00224	SUBL3	PTR, EOS, STR_LEN	0789		
	AB	68		57	28	00228	MOVC3	STR_LEN, (PTR), 1(PTR)	0790		
		68	04	AE	90	0022D	MOVB	ENDCHAR, (PTR)	0791		
		58	01	A748	9E	00231	MOVAB	1(STR_LEN)[PTR], PTR	0792		
				46	11	00236	BRB	24\$	0786		
		57		5A	C3	00238	SUBL3	STR_PTR, EOS, STR_LEN	0796		
			04	AE	90	0023C	MOVB	ENDCHAR, (PTR)+	0797		
		88		57	C0	00240	ADDL2	STR_LEN, PTR	0798		
		58		39	11	00243	BRB	24\$	0763		
		6A		2C	3A	00245	LOCC	#44, STR_LEN, (STR_PTR)	0809		
				02	12	00249	BNEQ	23\$			
				51	D4	0024B	CLRL	R1			
		6E		51	D0	0024D	MOVL	R1, STATUS			
		50	00000000G	00	9A	00250	MOVZBL	FILE_NAM+57, R0	0813		
		5A	01	A049	9E	00257	MOVAB	1(R0)[R9], STR_PTR			
07	50	56		5A	C3	0025C	SUBL3	STR_PTR, EOS, R0	0814		
	AA	6A		50	28	00260	MOVC3	R0, (STR_PTR), 7(STR_PTR)			
		5B		53	D0	00265	MOVL	R3, TEMP			
	6A	00000000'		06	28	00268	MOVC3	#6, P.AAS, (STR_PTR)	0815		
				53	D0	00270	MOVL	R3, STR_PTR			
				6A	04	AE	90	00273	MOVB	ENDCHAR, (STR_PTR)	0816
				6E	D5	00277	TSTL	STATUS	0821		
				05	12	00279	BNEQ	25\$			
		58		5B	D0	0027B	MOVL	TEMP, PTR			
				70	11	0027E	BRB	28\$			
24	00	6E		00	2C	00280	MOVC5	#0, (SP), #0, #36, TPARSE_BLOCK	0830		
				10	AE	00285					
		10		08	D0	00287	MOVL	#8, TPARSE_BLOCK	0831		
18	AE	56		5A	C3	0028B	SUBL3	STR_PTR, EOS, TPARSE_BLOCK+8	0833		
		53		AA	9E	00290	MOVAB	7(RT0), R3	0834		
		1C		53	D0	00294	MOVL	R3, TPARSE_BLOCK+12			
				00000000'	EF	9F	00298	PUSHAB	DIR_KEYS	0835	
				00000000'	EF	9F	0029E	PUSHAB	DIR_STATE		
				18	AE	9F	002A4	PUSHAB	TPARSE_BLOCK		
		00000000G		00	03	FB	002A7	CALLS	#3, LIB\$TPARSE		
				6E	D0	002AE	MOVL	R0, STATUS			
		08		6E	E9	002B1	BLBC	STATUS, 26\$		0842	
		0C		06	B0	002B4	MOVW	#6, TEMP_DESC		0843	
				53	D0	002B8	MOVL	R3, TEMP_DESC+4		0848	
				00000000G	00	DD	002BC	PUSHL	MEMBER	0847	
				00000000G	00	DD	002C2	PUSHL	GROUP	0844	
				10	AE	9F	002C8	PUSHAB	TEMP_DESC		



		14	AE 9F 002CB	PUSHAB	TEMP DESC	
		00000000C'	EF 9F 002CE	PUSHAB	P.AAT	
00000000G	00		05 FB 002D4	CALLS	#5, SYSS\$FA0	
	6E		50 D0 002DB	MOVL	R0, STATUS	
	0B		6E E8 002DE	BLBS	STATUS, 27\$	
			6E DD 002E1	PUSHL	STATUS	0849
00000000G	00		01 FB 002E3	CALLS	#1, LIB\$STOP	
			04 11 002EA	BRB	28\$	
	58	0E	AA 9E 002EC	MOVAB	14(R10), PTR	0850
	88	00000000'	EF D0 002F0	MOVL	P.AAV, (PTR)+	0854
00000000G	00		59 83 002F7	SUBB3	R9, PTR, FILE_FAB+52	0855
	58		59 D0 002FF	MOVL	R9, FILE_FAB+44	0856
00000000G	00		01 D0 00306	MOVL	#1, R0	0858
	50		04 00309	RET		
			50 D4 0030A	CLRL	R0	0859
			04 0030C	RET		

; Routine Size: 781 bytes, Routine Base: \$CODE\$ + 03CF

: 868 0860 1 END  
: 869 0861 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	100	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
_LIB\$KEYOS	0	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
_LIB\$STATES	16	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
\$CODES	1756	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALI (2)
\$PLITS	208	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Symbols		Percent	Pages Mapped	Processing Time
	Total	Loaded			
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	109	0	1000	00:01.7
_\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	18	42	14	00:00.1

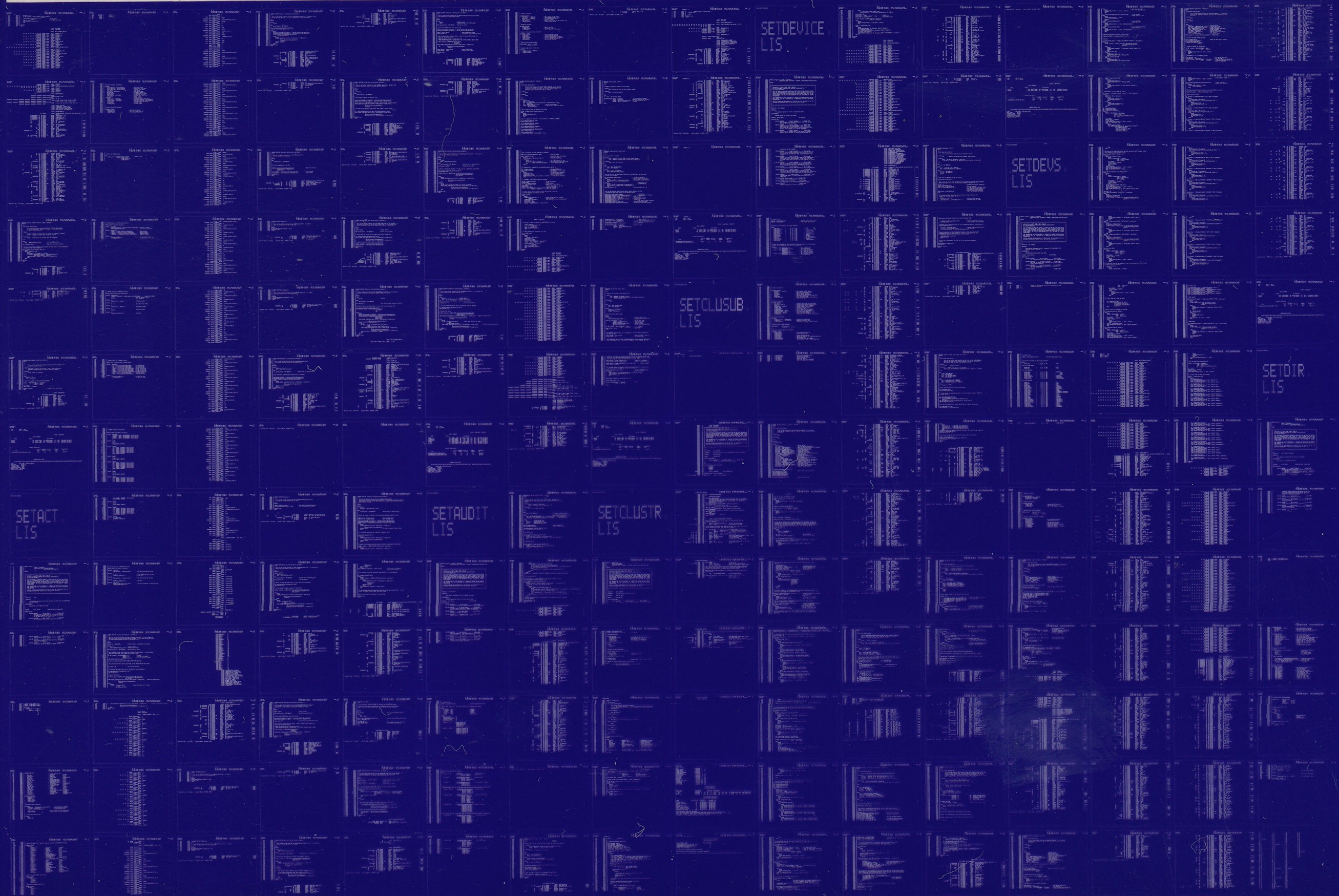
COMMAND QUALIFIERS

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

: Size: 1756 code + 324 data bytes  
 : Run Time: 00:32.7  
 : Elapsed Time: 01:40.0  
 : Lines/CPU Min: 1581  
 : Lexemes/CPU-Min: 25423  
 : Memory Used: 274 pages  
 : Compilation Complete

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

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY



0053 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 144 terminal windows, arranged in 12 rows and 12 columns. Each window shows a different terminal session on a VAX/VMS system. The sessions are characterized by text-based output, including command prompts and system responses. Several windows are clearly legible and show the following commands and their outputs:

- SETFILE LIS**: Shows a list of files and their attributes.
- SETPOMESS LIS**: Shows system messages or logs.
- SETP001SP LIS**: Shows system status or configuration.
- SETMISC LIS**: Shows miscellaneous system information.
- SETPRO LIS**: Shows process-related information.

The remaining windows in the grid show similar text-based data, though they are less legible due to the image quality and the density of the grid. The overall appearance is that of a multi-user terminal session on a VAX/VMS system.