

IIIIIIIIII	NNN	NNN	SSSSSSSSSSSS	TTTTTTTTTTTTTT	AAAAAAAAAA	LLL	
IIIIIIIIII	NNN	NNN	SSSSSSSSSSSS	TTTTTTTTTTTTTT	AAAAAAAAAA	LLL	
IIIIIIIIII	NNN	NNN	SSSSSSSSSSSS	TTTTTTTTTTTTTT	AAAAAAAAAA	LLL	
III	NNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNNNNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNNNNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNNNNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNN	NNN	NNN	SSSSSSSSS	AAA	AAA	LLL
III	NNN	NNN	NNN	SSSSSSSSS	AAA	AAA	LLL
III	NNN	NNN	NNN	SSSSSSSSS	AAA	AAA	LLL
III	NNN	NNNNNN	SSS	TTT	AAAAAAAAAAAAA	LLL	
III	NNN	NNNNNN	SSS	TTT	AAAAAAAAAAAAA	LLL	
III	NNN	NNNNNN	SSS	TTT	AAAAAAAAAAAAA	LLL	
III	NNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNN	NNN	SSS	TTT	AAA	AAA	LLL
III	NNN	NNN	SSS	TTT	AAA	AAA	LLL
IIIIIIIIII	NNN	NNN	SSSSSSSSSSSS	TTT	AAA	AAA	LLLLLLLLLLLLLLLL
IIIIIIIIII	NNN	NNN	SSSSSSSSSSSS	TTT	AAA	AAA	LLLLLLLLLLLLLLLL
IIIIIIIIII	NNN	NNN	SSSSSSSSSSSS	TTT	AAA	AAA	LLLLLLLLLLLLLLLL

```
IIIIII  NN  NN  SSSSSSSS  PPPPPPPP  RRRRRRRR  SSSSSSSS  000000  LL  DDDDDDDD
IIIIII  NN  NN  SSSSSSSS  PPPPPPPP  RRRRRRRR  SSSSSSSS  000000  LL  DDDDDDDD
  II    NN  NN  SS        PP        PP  RR        RR  SS        00        00  LL  DD        DD
  II    NN  NN  SS        PP        PP  RR        RR  SS        00        00  LL  DD        DD
  II    NNNN  NN  SS        PP        PP  RR        RR  SS        00        00  LL  DD        DD
  II    NNNN  NN  SS        PP        PP  RR        RR  SS        00        00  LL  DD        DD
  II    NN  NN  SSSSSS  PPPPPPPP  RRRRRRRR  SSSSSS  00        00  LL  DD        DD
  II    NN  NN  SSSSSS  PPPPPPPP  RRRRRRRR  SSSSSS  00        00  LL  DD        DD
  II    NN  NNNN  SS        PP        PP  RR  RR  SS        00        00  LL  DD        DD
  II    NN  NNNN  SS        PP        PP  RR  RR  SS        00        00  LL  DD        DD
  II    NN  NN  SS        PP        PP  RR  RR  SS        00        00  LL  DD        DD
  II    NN  NN  SS        PP        PP  RR  RR  SS        00        00  LL  DD        DD
IIIIII  NN  NN  SSSSSSSS  PP        PP  RR  RR  SSSSSSSS  000000  LLLLLLLLLL  DDDDDDDD
IIIIII  NN  NN  SSSSSSSS  PP        PP  RR  RR  SSSSSSSS  000000  LLLLLLLLLL  DDDDDDDD
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS
```



```
1 0001 0 MODULE INSPRSOLD (
2 0002 0 IDENT = 'V04-000' ; Parse old form of commands
3 0003 0 ADDRESSING_MODE(EXTERNAL = GENERAL)
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: Install
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 This module controls command parsing and processing.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 VAX/VMS operating system.
41 0041 1
42 0042 1 AUTHOR: Bob Grosso, June 1981
43 0043 1
44 0044 1 Modified by:
45 0045 1
46 0046 1 V03-007 MSH0061 Michael S. Harvey 5-Jul-1984
47 0047 1 Add EXEONLY support.
48 0048 1
49 0049 1 V03-006 MSH0055 Michael S. Harvey 31-May-1984
50 0050 1 Prevent infinite loops on command input errors.
51 0051 1
52 0052 1 V03-005 MSH0015 Michael S. Harvey 6-Mar-1984
53 0053 1 Clarify error message for /REPLACE command.
54 0054 1
55 0055 1 V03-004 BLS0256 Benn Schreiber 27-Dec-1983
56 0056 1 Remove conflicting qualifier checking, now done in cld.
57 0057 1 Allow /DELETE even if the file has been deleted.
```

INSPRSOLD
V04-000

L 16
16-Sep-1984 01:47:51 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:35:40 [INSTAL.SRC]INSPRSOLD.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	BLS0245	Benn Schreiber	27-Oct-1983
	Correct severity of FAIL message for cases where WARNING is appropriate.		
V03-002	RPG0002	Bob Grosso	13-Sep-1983
	Re-activate header resident. Enable /NOPURGE for old interface.		
V03-001	RPG0001	Bob Grosso	29-Aug-1983
	Print help from help library instead of program text.		

Declarations

```

: 71      0070 1 %SBTTL 'Declarations';
: 72      0071 1
: 73      0072 1
: 74      0073 1 | Include files
: 75      0074 1 |
: 76      0075 1
: 77      0076 1 LIBRARY 'SYS$LIBRARY:LIB';          ! VAX/VMS system definitions
: 78      0077 1
: 79      0078 1 REQUIRE 'SRC$:INSPREFIX.REQ';
: 80      0220 1 REQUIRE 'LIB$:INSDEF.R32';
: 81      0279 1
: 82      0280 1
: 83      0281 1 |
: 84      0282 1 | Table of contents
: 85      0283 1 |
: 86      0284 1
: 87      0285 1 FORWARD ROUTINE
: 88      0286 1     PROCESS COMMAND,                ! execute the command line
: 89      0287 1     DISPATCH,                        ! check for conflicting quals and
: 90      0288 1                                     ! dispatch to appropriate routine
: 91      0289 1     HELP;                            ! print interactive help
: 92      0290 1
: 93      0291 1 EXTERNAL
: 94      0292 1     INSOLDCMD;                        ! Tables describing the old Install commands
: 95      0293 1
: 96      0294 1 |
: 97      0295 1 | External routines
: 98      0296 1 |
: 99      0297 1
: 100     0298 1 EXTERNAL ROUTINE
: 101     0299 1     ! Module INSMAN
: 102     0300 1     INSSOPEN_FILE,                    ! handle file open processing
: 103     0301 1     INSSCLOSE_FILE,                  ! handle file close
: 104     0302 1     INSSCHECK_PRIV,                  ! enforce privilege check in case INSTALL is installed
: 105     0303 1
: 106     0304 1     INSSCREATE,                      ! perform CREATE command
: 107     0305 1     INSSDELETE,                      ! perform DELETE command
: 108     0306 1     INSSGLOBAL,                      ! list global sections
: 109     0307 1     INSSLIST;                        ! list known file entries
: 110     0308 1
: 111     0309 1 EXTERNAL ROUTINE
: 112     0310 1     CLISDCL_PARSE,                    ! Parse DCL command
: 113     0311 1     CLISPRESENT,                     ! Check if entity present
: 114     0312 1     CLISGET_VALUE,                   ! Get value from command line
: 115     0313 1     LBR$OUTPUT_HELP,                 ! print interactive help
: 116     0314 1     LIB$GET_INPUT,                   ! Get subcommands
: 117     0315 1     LIB$PUT_OUTPUT,                   ! echo command line, and print help
: 118     0316 1     PRVS$SETPRIV,                    ! set privilege bits given ASCII string
: 119     0317 1     STR$COMPARE,                     ! compare two strings
: 120     0318 1     STR$CONCAT,                      ! concatenate strings
: 121     0319 1     STR$COPY_DX,                     ! Copy strings by descriptor
: 122     0320 1     STR$POSITION,
: 123     0321 1     STR$RIGHT,
: 124     0322 1     STR$UPCASE;
: 125     0323 1
: 126     0324 1 EXTERNAL LITERAL
: 127     0325 1     CLIS_NEGATED,                    ! Qualifier was explicitly negated
```



```
128 0326 1 PRV$_INVNAM,  
129 0327 1 PRV$_NOTUNQ;  
130 0328 1  
131 0329 1 EXTERNAL LITERAL  
132 0330 1 INSS_EXISTS,  
133 0331 1 INSS_FAIL,  
134 0332 1 INSS_FAILED,  
135 0333 1 INSS_HELP,  
136 0334 1 INSS_INTRNLERR,  
137 0335 1 INSS_INVPRVNAM,  
138 0336 1 INSS_NOKFEFND,  
139 0337 1 INSS_NOLOAD,  
140 0338 1 INSS_PRVNOTUNQ,  
141 0339 1 INSS_REMOVED;  
142 0340 1  
143 0341 1 EXTERNAL  
144 0342 1 INSSL_INTRNLERR,  
145 0343 1 INSSG_KFENAM,  
146 0344 1 INSSG_KFECHAN,  
147 0345 1 INSSGL_CTLMSK : $BBLOCK,  
148 0346 1 INSSGL_REPLACE_MSK: $BBLOCK,  
149 0347 1 INSSGQ_KFERNS :  
150 0348 1 $BBLOCK,  
151 0349 1 INSSGL_KFEADR,  
152 0350 1 INSSGQ_KFEPRIVS : $BBLOCK,  
153 0351 1 INSSGQ_FILDSC :  
154 0352 1 $BBLOCK,  
155 0353 1 INSSG_KFEFAB : $BBLOCK,  
156 0354 1  
157 0355 1 CTL$GQ_PROCPRIV : $BBLOCK;  
158 0356 1  
159 0357 1 OWN  
160 0358 1 INPUT_DESC : DYN_DESC_DECL;  
161 0359 1  
162 0360 1 BIND  
163 0361 1 EXIT_DSC = $DESCRIPTOR ('EXIT'),  
164 0362 1 FILE_SPEC_DSC = $DESCRIPTOR ('FILE_SPEC'),  
165 0363 1 GLOBAL_DSC = $DESCRIPTOR ('GLOBAL'),  
166 0364 1 LIST_DSC = $DESCRIPTOR ('LIST'),  
167 0365 1 PRIVILEGED_DSC = $DESCRIPTOR ('PRIVILEGED'),  
168 0366 1 CREATE_DSC = $DESCRIPTOR ('CREATE'),  
169 0367 1 REPLACE_DSC = $DESCRIPTOR ('REPLACE'),  
170 0368 1 DELETE_DSC = $DESCRIPTOR ('DELETE'),  
171 0369 1 SLASH_HEL_DSC = $DESCRIPTOR ('/HEL'),  
172 0370 1 SLASH_HELP_DSC = $DESCRIPTOR ('/HELP'),  
173 0371 1 HEL_DSC = $DESCRIPTOR ('HEL'),  
174 0372 1 HELP_DSC = $DESCRIPTOR ('HELP');  
175 0373 1
```

! Invalid privilege name
! Privilege name not unique

! Known File Entry already exists
! Unable to <OPERATION> <FILENAME>
! Failed to process request for <FILENAME>
! Enter /HELP for INSTALL help
! Internal Error: error detail
! Invalid privilege name
! Known File not found
! Network and sequentially loaded files are not installable
! Privilege name not unique
! Known entry was removed

! Store internal error descriptor
! file name block declaration
! channel for open known file
! Control flags
! Control flags saved for REPLACE
! buffer for resultant name

! id of file to be listed
! Quadword privilege mask
! descriptor to get filename returned
! from CLI.
! file access block declaration

! privilege mask of user

! command string input


```
177 0374 1 %SBTTL 'INSPARSE_OLD_CMD';
178 0375 1
179 0376 1 GLOBAL ROUTINE INSPARSE_OLD_CMD ( PARSE_CMD_DESC, INTERACTIVE ) =
180 0377 2 BEGIN
181 0378 2 |+++
182 0379 2 |
183 0380 2 | Command loop. If not interactive, execute command and exit, else
184 0381 2 | print "INSTALL>", read in command, parse command and execute command.
185 0382 2 | Exit loop for ^Z and /EXIT
186 0383 2 |
187 0384 2 |---
188 0385 2 MAP
189 0386 2 PARSE_CMD_DESC : REF $BLOCK;
190 0387 2
191 0388 2 DO
192 0389 2 BEGIN
193 0390 2 LOCAL
194 0391 2 HLP_PTR,
195 0392 2 HLP_LEN,
196 0393 2 STATUS;
197 0394 2
198 0395 2 IF .INTERACTIVE
199 0396 2 THEN
200 0397 2 BEGIN
201 0398 2 LOCAL
202 0399 2 COPY_DESC : $BLOCK [DSC$C_S_BLN],
203 0400 2 DASH_PTR, ! Locate the '-' continuation character
204 0401 2 LAST_WAS_CONT; ! Last command line was continued
205 0402 2
206 0403 2 ! print the prompt and get input line
207 0404 2 STATUS = LIB$GET INPUT
208 0405 2 ( INPUT_DESC, $DESCRIPTOR('INSTALL>') );
209 0406 2 IF .STATUS EQL RMSS_EOF ! EXIT on CNTRL Z
210 0407 2 THEN EXITLOOP;
211 0408 2 IF NOT .STATUS
212 0409 2 THEN
213 0410 2 BEGIN
214 0411 2 SIGNAL (.STATUS);
215 0412 2 RETURN .STATUS;
216 0413 2 END;
217 0414 2
218 0415 2 DASH_PTR = CH$FIND CH( .INPUT_DESC [DSC$W_LENGTH],
219 0416 2 .INPUT_DESC [DSC$A_POINTER], %C'-');
220 0417 2 IF NOT CH$FAIL (.DASH_PTR)
221 0418 2 THEN
222 0419 2 BEGIN ! Line continued
223 0420 2 LAST_WAS_CONT = TRUE;
224 0421 2 COPY_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
225 0422 2 COPY_DESC [DSC$W_LENGTH] = .DASH_PTR = .INPUT_DESC [DSC$A_POINTER];
226 0423 2 COPY_DESC [DSC$A_POINTER] = .INPUT_DESC [DSC$A_POINTER];
227 0424 2 STR$COPY DX ( .PARSE_CMD_DESC, COPY_DESC ); ! copy dst from src
228 0425 2 WHILE .LAST_WAS_CONT DO
229 0426 2 |
230 0427 2 | while last was continued, input next line,
231 0428 2 | and concatenate with previous.
232 0429 2 |
233 0430 2 BEGIN
```



```
234 0431 6 STATUS = LIB$GET INPUT( INPUT_DESC, $DESCRIPTOR ('_> ')); ! get input line
235 0432 6 IF .STATUS EQL RMSS_EOF
236 0433 6 THEN
237 0434 7 BEGIN
238 0435 7 INTERACTIVE = FALSE;
239 0436 7 EXITLOOP;
240 0437 6 END;
241 0438 6
242 0439 6 COPY_DESC [DSC$A_POINTER] = .INPUT_DESC [DSC$A_POINTER];
243 0440 6
244 0441 6 DASH_PTR = CH$FIND_CH( .INPUT_DESC [DSC$W_LENGTH], .INPUT_DESC [DSC$A_POINTER], %c'-' );
245 0442 6 IF CH$FAIL (.DASH_PTR)
246 0443 6 THEN
247 0444 7 BEGIN
248 0445 7 LAST_WAS_CONT = FALSE;
249 0446 7 COPY_DESC [DSC$W_LENGTH] = .INPUT_DESC [DSC$W_LENGTH];
250 0447 7 END
251 0448 6 ELSE
252 0449 7 BEGIN
253 0450 7 COPY_DESC [DSC$W_LENGTH] = .DASH_PTR - .INPUT_DESC [DSC$A_POINTER];
254 0451 6 END;
255 0452 6 STR$CONCAT (.PARSE_CMD_DESC, .PARSE_CMD_DESC, COPY_DESC );
256 0453 5 END; ! While reading lines following continuation lines
257 0454 5
258 0455 5 IF NOT .INTERACTIVE THEN EXITLOOP; ! ^Z entered on continuation line
259 0456 5 STR$UPCASE ( .PARSE_CMD_DESC, .PARSE_CMD_DESC );
260 0457 5 END ! If line continued
261 0458 4 ELSE
262 0459 4 STR$UPCASE ( .PARSE_CMD_DESC, INPUT_DESC ); ! Ucase and copy to PARSE_CMD_DESC
263 0460 3 END; ! If interactive
264 0461 3
265 0462 3
266 0463 3
267 0464 3
268 0465 3 Parse command if there is something on the command line
269 0466 3
270 0467 3
271 0468 3 IF ( .PARSE_CMD_DESC [DSC$W_LENGTH] NEQ 0 ) AND ! If there is something on the line
272 0469 3 NOT CH$FAIL( CH$FIND_NOT_CH ! and there is something besides blanks
273 0470 3 ( .PARSE_CMD_DESC [DSC$W_LENGTH],
274 0471 3 .PARSE_CMD_DESC [DSC$A_POINTER], %c' '))
275 0472 3 THEN
276 0473 4 BEGIN
277 0474 4
278 0475 4
279 0476 4
280 0477 4
281 0478 4
282 0479 4
283 0480 4
284 0481 4 HLP_PTR = STR$POSITION ( .PARSE_CMD_DESC, SLASH_HELP_DSC); ! Look for "/HELP" on comman
285 0482 4 HLP_LEN = 5; ! If not found, look for "/H
286 0483 4 IF .HLP_PTR EQL 0
287 0484 4 THEN
288 0485 5 BEGIN
289 0486 5 HLP_PTR = STR$POSITION ( .PARSE_CMD_DESC, SLASH_HEL_DSC);
290 0487 5 HLP_LEN = 4;
```



```
291 0488 4      END;  
292 0489 4      IF .HLP_PTR NEQ 0  
293 0490 4      THEN  
294 0491 4          Help was requested, treat entire line as help request  
295 0492 4      BEGIN  
296 0493 4          replace "/HEL" or "/HELP" with "INSTALL" since lbr$output_help  
297 0494 5          needs to see the topic keyword.  
298 0495 5          Then call the help routine.  
299 0496 5      STR$RIGHT ( .PARSE_CMD_DESC, .PARSE_CMD_DESC, %REF (.HLP_PTR+.HLP_LEN) );  
300 0497 5      STR$CONCAT (.PARSE_CMD_DESC, $DESCRIPTOR ('INSTALL '), .PARSE_CMD_DESC);  
301 0498 5          ! Stick "INSTALL" on front of help line  
302 0499 5      REPORT( HELP ( .PARSE_CMD_DESC ) );  
303 0500 5      END  
304 0501 5          ! Help was requested  
305 0502 5      ELSE  
306 0503 5      BEGIN  
307 0504 5          ! Command was something other than /HELP  
308 0505 5      BEGIN  
309 0506 5          !+++  
310 0507 5          No help requested, proceed to parsing command line  
311 0508 5          Must concatenate command line to "INSTALL" string for  
312 0509 5          cli$dcl_parse to execute.  
313 0510 5      ---  
314 0511 5      STR$CONCAT (.PARSE_CMD_DESC, $DESCRIPTOR ('INSTALL '), .PARSE_CMD_DESC); ! Stick "INSTALL" on front  
315 0512 5      STATUS = CLISDCL_PARSE( .PARSE_CMD_DESC, INSOLD_CMD ); ! parse the command line  
316 0513 5      IF .STATUS  
317 0514 5      THEN  
318 0515 5      BEGIN  
319 0516 5          IF CLIS$PRESENT( EXIT_DSC ) ! If /EXIT, then exit program  
320 0517 5          THEN EXITLOOP;  
321 0518 5      !+++  
322 0519 5          Set flags for the qualifiers present in command line  
323 0520 5      ---  
324 0521 6      INSSGL_CTLMSK = 0; ! clear flags  
325 0522 6      INSSL_INTRNLERR = 0; ! clear internal error descriptor  
326 0523 6      INSTALL control flags  
327 0524 6      INSSGL_CTLMSK [INSSV_FILSPC] = CLIS$PRESENT( FILE_SPEC_DSC );  
328 0525 6      INSSGL_CTLMSK [INSSV_CREATE] = CLIS$PRESENT( CREATE_DSC );  
329 0526 6      INSSGL_CTLMSK [INSSV_DELETE] = CLIS$PRESENT( DELETE_DSC );  
330 0527 6      INSSGL_CTLMSK [INSSV_REPLACE] = CLIS$PRESENT( REPLACE_DSC );  
331 0528 6      INSSGL_CTLMSK [INSSV_LIST] = CLIS$PRESENT( LIST_DSC );  
332 0529 6      INSSGL_CTLMSK [INSSV_FULL] = CLIS$PRESENT( %ASCID 'FULL' );  
333 0530 6      INSSGL_CTLMSK [INSSV_GLOBAL] = CLIS$PRESENT( GLOBAL_DSC );  
334 0531 6      INSSGL_CTLMSK [INSSV_HELP] = CLIS$PRESENT( %ASCID 'HELP' );  
335 0532 6  
336 0533 6  
337 0534 6  
338 0535 6  
339 0536 6  
340 0537 6  
341 0538 6  
342 0539 6  
343 0540 6  
344 0541 6  
345 0542 6  
346 0543 6  
347 0544 6
```



```

: 348      0545 6      INSSGL-CTLMSK [INSSV_EXIT] = CLIPRESENT( %ASCID 'EXIT' );
: 349      0546 6      INSSGL-CTLMSK [INSSV_PROTECT] = CLIPRESENT( %ASCID 'PROTECTED' );
: 350      0547 6      INSSGL-CTLMSK [INSSV_OPEN] = CLIPRESENT( %ASCID 'OPEN' );
: 351      0548 6      INSSGL-CTLMSK [INSSV_HDRRES] = CLIPRESENT( %ASCID 'HEADER_RESIDENT' );
: 352      0549 6      INSSGL-CTLMSK [INSSV_PRIV] = CLIPRESENT( PRIVILEGED_DSC );
: 353      0550 6      INSSGL-CTLMSK [INSSV_SHARED] = CLIPRESENT( %ASCID 'SHARED' );
: 354      0551 6      INSSGL-CTLMSK [INSSV_WRITABLE] = CLIPRESENT( %ASCID 'WRITEABLE' );
: 355      0552 6      INSSGL-CTLMSK [INSSV_NOPURGE] = CLIPRESENT( %ASCID 'NOPURGE' );
: 356      0553 6      INSSGL-CTLMSK [INSSV_EXEONLY] = CLIPRESENT( %ASCID 'EXECUTE_ONLY' );
: 357      0554 6
: 358      0555 6      !
: 359      0556 6      !       execute the command
: 360      0557 6      !
: 361      0558 6      REPORT (PROCESS_COMMAND( ) );
: 362      0559 6      END;                               ! If it parsed
: 363      0560 5
: 364      0561 4      END;                               ! Else command was something other than /HELP
: 365      0562 3      END;                               ! If the command line was non-blank
: 366      0563 3      END                               ! while
: 367      0564 2      WHILE .INTERACTIVE;
: 368      0565 2      RETURN TRUE;
: 369      0566 1      END;                               ! Routine Parse_old_cmd
```

```

.TITLE INSPRSOLD
.IDENT \V04-000\

.PSECT $SPLIT$,NOWRT,NOEXE,2
```

```

      54 49 58 45 00000 P.AAB: .ASCII \EXIT\
      00000004 00004 P.AAA: .LONG 4
      00000000 00008 P.AAA: .ADDRESS P.AAB
43 45 50 53 5F 45 4C 49 46 0000C P.AAD: .ASCII \FILE_SPEC\
      00015 P.AAD: .BLKB 3
      00000009 00018 P.AAC: .LONG 9
      00000000 0001C P.AAC: .ADDRESS P.AAD
      4C 41 42 4F 4C 47 00020 P.AAF: .ASCII \GLOBAL\
      00026 P.AAF: .BLKB 2
      00000006 00028 P.AAE: .LONG 6
      00000000 0002C P.AAE: .ADDRESS P.AAF
      54 53 49 4C 00030 P.AAH: .ASCII \LIST\
      00000004 00034 P.AAG: .LONG 4
      00000000 00038 P.AAG: .ADDRESS P.AAH
44 45 47 45 4C 49 56 49 52 50 0003C P.AAJ: .ASCII \PRIVILEGED\
      00046 P.AAJ: .BLKB 2
      0000000A 00048 P.AAI: .LONG 10
      00000000 0004C P.AAI: .ADDRESS P.AAJ
      45 54 41 45 52 43 00050 P.AAL: .ASCII \CREATE\
      00056 P.AAL: .BLKB 2
      00000006 00058 P.AAK: .LONG 6
      00000000 0005C P.AAK: .ADDRESS P.AAL
45 43 41 4C 50 45 52 00060 P.AAN: .ASCII \REPLACE\
      00067 P.AAN: .BLKB 1
      00000007 00068 P.AAM: .LONG 7
      00000000 0006C P.AAM: .ADDRESS P.AAN
45 54 45 4C 45 44 00070 P.AAP: .ASCII \DELETE\
      00076 P.AAP: .BLKB 2
```



```
00000006 00078 P.AAO: .LONG 6
00000000 0007C .ADDRESS P.AAP
4C 45 48 2F 00080 P.AAR: .ASCII \HEL\
00000004 00084 P.AAQ: .LONG 4
00000000 00088 .ADDRESS P.AAR
50 4C 45 48 2F 0008C P.AAT: .ASCII \HELP\
00091 .BLKB 3
00000005 00094 P.AAS: .LONG 5
00000000 00098 .ADDRESS P.AAT
4C 45 48 0009C P.AAV: .ASCII \HEL\
0009F .BLKB 1
00000003 000A0 P.AAU: .LONG 3
00000000 000A4 .ADDRESS P.AAV
50 4C 45 48 000A8 P.AAX: .ASCII \HELP\
00000004 000AC P.AAW: .LONG 4
00000000 000B0 .ADDRESS P.AAX
20 3E 4C 4C 41 54 53 4E 49 000B4 P.AAZ: .ASCII \INSTALL> \
000BD .BLKB 3
00000009 000C0 P.AAY: .LONG 9
00000000 000C4 .ADDRESS P.AAZ
20 3E 5F 000C8 P.ABB: .ASCII \> \
000CB .BLKB 1
00000003 000CC P.ABA: .LONG 3
00000000 000D0 .ADDRESS P.ABB
20 4C 4C 41 54 53 4E 49 000D4 P.ABD: .ASCII \INSTALL \
00000008 000DC P.ABC: .LONG 8
00000000 000E0 .ADDRESS P.ABD
20 4C 4C 41 54 53 4E 49 000E4 P.ABF: .ASCII \INSTALL \
00000008 000EC P.ABE: .LONG 8
00000000 000F0 .ADDRESS P.ABF
4C 4C 55 46 000F4 P.ABH: .ASCII \FULL\
010E0004 000F8 P.ABG: .LONG 17694724
00000000 000FC .ADDRESS P.ABH
50 4C 45 48 00100 P.ABJ: .ASCII \HELP\
010E0004 00104 P.ABI: .LONG 17694724
00000000 00108 .ADDRESS P.ABJ
54 49 58 45 0010C P.ABL: .ASCII \EXIT\
010E0004 00110 P.ABK: .LONG 17694724
00000000 00114 .ADDRESS P.ABL
00 00 00 44 45 54 43 45 54 4F 52 50 00118 P.ABN: .ASCII \PROTECTED\<0><0><0>
010E0009 00124 P.ABM: .LONG 17694729
00000000 00128 .ADDRESS P.ABN
4E 45 50 4F 0012C P.ABP: .ASCII \OPEN\
010E0004 00130 P.ABO: .LONG 17694724
00000000 00134 .ADDRESS P.ABP
54 4E 45 44 49 53 45 52 5F 52 45 44 41 45 48 00138 P.ABR: .ASCII \HEADER_RESIDENT\<0>
00 00147
010E000F 00148 P.ABQ: .LONG 17694735
00000000 0014C .ADDRESS P.ABR
00 00 44 45 52 41 48 53 00150 P.ABT: .ASCII \SHARED\<0><0>
010E0006 00158 P.ABS: .LONG 17694726
00000000 0015C .ADDRESS P.ABT
00 00 00 45 4C 42 41 45 54 49 52 57 00160 P.ABV: .ASCII \WRITEABLE\<0><0><0>
010E0009 0016C P.ABU: .LONG 17694729
00000000 00170 .ADDRESS P.ABV
00 45 47 52 55 50 4F 4E 00174 P.ABX: .ASCII \NOPURGE\<0>
010E0007 0017C P.ABW: .LONG 17694727
```


59 4C 4E 4F 5F 45 54 55 43 45 00000000' 00180
58 45 00184 P.ABZ: .ADDRESS P.ABX
010E000C 00190 P.ABY: .ASCII \EXECUTE_ONLY\
00000000' 00194 .LONG 17694732
.ADDRESS P.ABZ

.PSECT \$OWNS\$,NOEXE,2

00# 00000 INPUT_DESC:
02 0E 00002 .BYTE 0[2]
00004 .BYTE 14, 2
.BLKB 4

EXIT_DSC= P.AAA
FILE_SPEC_DSC= P.AAC
GLOBAL_DSC= P.AAE
LIST_DSC= P.AAG
PRIVILEGED_DSC= P.AAI
CREATE_DSC= P.AAK
REPLACE_DSC= P.AAM
DELETE_DSC= P.AAO
SLASH_HELP_DSC= P.AAQ
SLASH_HELP_DSC= P.AAS
HEL_DSC= P.AAU
HELP_DSC= P.AAW

.EXTRN INSOLDCMD, INSSOPEN FILE
.EXTRN INSSCLOSE FILE, INSSCHECK_PRIV
.EXTRN INSSCREATE, INSSDELETE
.EXTRN INSSGLOBAL, INSSLIST
.EXTRN CLISDCL_PARSE, CLISPRESENT
.EXTRN CLISGET_VALUE, LBR\$OUTPUT_HELP
.EXTRN LIB\$GET_INPUT, LIB\$PUT_OUTPUT
.EXTRN PRV\$SETPRIV, STR\$COMPARE
.EXTRN STR\$CONCAT, STR\$COPY DX
.EXTRN STR\$POSITION, STR\$RIGHT
.EXTRN STR\$UPCASE, CLIS NEGATED
.EXTRN PRV\$ INVNAM, PRV\$ NOTUNQ
.EXTRN INSS_EXISTS, INSS_FAIL
.EXTRN INSS_FAILED, INSS_HELP
.EXTRN INSS_INTRNLERR, INSS INVPRV NAM
.EXTRN INSS_NOKFEFND, INSS_NOLOAD
.EXTRN INSS_PRVNOTUNQ, INSS_REMOVED
.EXTRN INSS_INTRNLERR
.EXTRN INSSG_KFENAM, INSSGL_KFECHAN
.EXTRN INSSGL_CTLMSK, INSSGL_REPLACE_MSK
.EXTRN INSSGQ_KFERNS, INSSGL_KFEADR
.EXTRN INSSGQ_KFEPRIVS
.EXTRN INSSGQ_FILDSC, INSSG_KFEFAB
.EXTRN CTL\$GQ_PROCPRIV

.PSECT \$CODE\$,NOWRT,2

OFFC 00000
5B 0000' CF 9E 00002
5A 0000' CF 9E 00007
59 00000000G 00 9E 0000C
58 00000000G 00 9E 00013

.ENTRY INSSPARSE_OLD_CMD, Save R2,R3,R4,R5,R6,R7,- ; 0376
R8,R9,R10,R11
MOVAB INPUT_DESC, R11
MOVAB P.AAY, R10
MOVAB INSSGL_CTLMSK, R9
MOVAB CLISPRESENT, R8

		5E		0C	C2	0001A	SUBL2	#12, SP		
		57		08	AC	D0	0001D	MOVL	INTERACTIVE, R7	0395
		52		04	AC	D0	00021	MOVL	PARSE_CMD_DESC, R2	0468
		03			57	E8	00025	BLBS	R7, 1\$	0395
					00C5	31	00028	BRW	14\$	
					5A	DD	0002B	PUSHL	R10	0405
					5B	DD	0002D	PUSHL	R11	
					02	FB	0002F	CALLS	#2, LIB\$GET_INPUT	
	00000000G	00			50	D0	00036	MOVL	R0, STATUS	
	0001827A	8F			55	D1	00039	CMPL	STATUS, #98938	0406
					03	12	00040	BNEQ	2\$	
					022F	31	00042	BRW	23\$	
		0D			55	E8	00045	BLBS	STATUS, 3\$	0408
					55	DD	00048	PUSHL	STATUS	0411
	00000000G	00			01	FB	0004A	CALLS	#1, LIB\$SIGNAL	
		50			55	D0	00051	MOVL	STATUS, R0	0412
						04	00054	RET		
		53		04	AB	D0	00055	MOVL	INPUT_DESC+4, R3	0416
63		6B			2D	3A	00059	LOCC	#45, INPUT_DESC, (R3)	0415
					02	12	0005D	BNEQ	4\$	
		54			51	D4	0005F	CLRL	R1	
		52			51	D0	00061	MOVL	R1, DASH_PTR	
				04	AC	D0	00064	MOVL	PARSE_CMD_DESC, R2	0424
					54	D5	00068	TSTL	DASH_PTR	0417
					79	13	0006A	BEQL	12\$	
		56			01	D0	0006C	MOVL	#1, LAST_WAS_CONT	0420
		AE	07		01	90	0006F	MOVB	#1, COPY_DESC+3	0421
04	AE	54			53	A3	00073	SUBW3	R3, DASH_PTR, COPY_DESC	0422
		AE	08		53	D0	00078	MOVL	R3, COPY_DESC+4	0423
				04	AE	9F	0007C	PUSHAB	COPY_DESC	0424
					52	DD	0007F	PUSHL	R2	
	00000000G	00			02	FB	00081	CALLS	#2, STR\$COPY_DX	
		4F			56	E9	00088	BLBC	LAST_WAS_CONT, 10\$	0425
				0C	AA	9F	0008B	PUSHAB	P.ABA	0431
					5B	DD	0008E	PUSHL	R11	
	00000000G	00			02	FB	00090	CALLS	#2, LIB\$GET_INPUT	
		55			50	D0	00097	MOVL	R0, STATUS	
	0001827A	8F			55	D1	0009A	CMPL	STATUS, #98938	0432
					05	12	000A1	BNEQ	6\$	
				08	AC	D4	000A3	CLRL	INTERACTIVE	0435
					32	11	000A6	BRB	10\$	0434
		53		04	AB	D0	000A8	MOVL	INPUT_DESC+4, R3	0439
		AE	08		53	D0	000AC	MOVL	R3, COPY_DESC+4	
63		6B			2D	3A	000B0	LOCC	#45, INPUT_DESC, (R3)	0441
					02	12	000B4	BNEQ	7\$	
		54			51	D4	000B6	CLRL	R1	
					51	D0	000B8	MOVL	R1, DASH_PTR	
					08	12	000BB	BNEQ	8\$	0442
					56	D4	000BD	CLRL	LAST_WAS_CONT	0445
		AE	04		6B	B0	000BF	MOVW	INPUT_DESC, COPY_DESC	0446
					05	11	000C3	BRB	9\$	0442
04	AE	54			53	A3	000C5	SUBW3	R3, DASH_PTR, COPY_DESC	0450
				04	AE	9F	000CA	PUSHAB	COPY_DESC	0452
					52	DD	000CD	PUSHL	R2	
					52	DD	000CF	PUSHL	R2	
	00000000G	00			03	FB	000D1	CALLS	#3, STR\$CONCAT	
					AE	11	000D8	BRB	5\$	0425

03	08	AC	E8	000DA	10\$:	BLBS	INTERACTIVE, 11\$	: 0455
	0193	31	000DE			BRW	23\$	: 0456
	52	DD	000E1	11\$:	PUSHL	R2		: 0459
	02	11	000E3		BRB	13\$		: 0468
	5B	DD	000E5	12\$:	PUSHL	R11		: 0470
	52	DD	000E7	13\$:	PUSHL	R2		: 0471
00000000G	00	02	FB	000E9	CALLS	#2, STR\$UPCASE		: 0481
	62	B5	000F0	14\$:	TSTW	(R2)		: 0482
04	B2	7F	13	000F2	BEQL	18\$		: 0483
	20	3B	000F4		SKPC	#32, (R2), @4(R2)		: 0486
	02	12	000F9		BNEQ	15\$		: 0487
	51	D4	000FB	15\$:	CLRL	R1		: 0489
	51	D5	000FD		TSTL	R1		: 0500
	72	13	000FF		BEQL	18\$		: 0501
	D4	AA	9F	00101	PUSHAB	SLASH_HELP_DSC		: 0503
	52	DD	00104		PUSHL	R2		: 0516
00000000G	00	02	FB	00106	CALLS	#2, STR\$POSITION		: 0517
53		05	D0	0010D	MOVL	#5, HLP_LEN		: 0519
	50	D5	00110		TSTL	HLP_PTR		: 0522
	0F	12	00112		BNEQ	16\$		: 0531
	C4	AA	9F	00114	PUSHAB	SLASH_HEL_DSC		: 0532
	52	DD	00117		PUSHL	R2		: 0536
00000000G	00	02	FB	00119	CALLS	#2, STR\$POSITION		: 0538
53		04	D0	00120	MOVL	#4, HLP_LEN		: 0531
	50	D5	00123	16\$:	TSTL	HLP_PTR		: 0532
	29	13	00125		BEQL	17\$		: 0536
6E	50	53	C1	00127	ADDL3	HLP_LEN, HLP_PTR, (SP)		: 0500
	4004	8F	BB	0012B	PUSHR	#^MZR2, SP>		: 0501
	52	DD	0012F		PUSHL	R2		: 0503
00000000G	00	03	FB	00131	CALLS	#3, STR\$RIGHT		: 0516
		52	DD	00138	PUSHL	R2		: 0517
	1C	AA	9F	0013A	PUSHAB	P.ABC		: 0519
	52	DD	0013D		PUSHL	R2		: 0522
00000000G	00	03	FB	0013F	CALLS	#3, STR\$CONCAT		: 0531
		52	DD	00146	PUSHL	R2		: 0532
0000V	CF	01	FB	00148	CALLS	#1, HELP		: 0536
	010E	31	0014D		BRW	21\$		: 0538
	52	DD	00150	17\$:	PUSHL	R2		: 0531
	2C	AA	9F	00152	PUSHAB	P.ABE		: 0532
	52	DD	00155		PUSHL	R2		: 0536
00000000G	00	03	FB	00157	CALLS	#3, STR\$CONCAT		: 0538
	00000000G	00	9F	0015E	PUSHAB	INSOLDCMD		: 0531
	52	DD	00164		PUSHL	R2		: 0532
00000000G	00	02	FB	00166	CALLS	#2, CLISDCL_PARSE		: 0536
55		50	D0	0016D	MOVL	R0, STATUS		: 0538
03		55	E8	00170	BLBS	STATUS, 19\$		: 0531
	00F4	31	00173	18\$:	BRW	22\$		: 0532
	FF44	CA	9F	00176	19\$:	PUSHAB	EXIT_DSC	: 0536
68		01	FB	0017A	CALLS	#1, CLISPRESENT		: 0538
03		50	E9	0017D	BLBC	R0, 20\$		: 0531
	00F1	31	00180		BRW	23\$		: 0532
	69	D4	00183	20\$:	CLRL	INSSGL_CTLMSK		: 0536
	00000000G	00	D4	00185	CLRL	INSSL_INTRNLERR		: 0538
	FF58	CA	9F	0018B	PUSHAB	FILE_SPEC_DSC		: 0531
68		01	FB	0018F	CALLS	#1, CLISPRESENT		: 0532
00		50	F0	00192	INSV	R0, #0, #1, INSSGL_CTLMSK		: 0536
	98	AA	9F	00197	PUSHAB	CREATE_DSC		: 0538

69	01	68	01	FB	0019A	CALLS	#1, CLISPRESENT		
		04	50	FO	0019D	INSV	R0, #4, #1, INSSGL_CTLMSK		
			B8	AA	9F 001A2	PUSHAB	DELETE DSC	0539	
69	01	68	01	FB	001A5	CALLS	#1, CLISPRESENT		
		06	50	FO	001A8	INSV	R0, #6, #1, INSSGL_CTLMSK		
			A8	AA	9F 001AD	PUSHAB	REPLACE DSC	0540	
69	01	68	01	FB	001B0	CALLS	#1, CLISPRESENT		
		05	50	FO	001B3	INSV	R0, #5, #1, INSSGL_CTLMSK		
			FF74	CA	9F 001B8	PUSHAB	LIST DSC	0541	
01	A9	68	01	FB	001BC	CALLS	#1, CLISPRESENT		
		01	50	FO	001BF	INSV	R0, #1, #1, INSSGL_CTLMSK+1		
			38	AA	9F 001C5	PUSHAB	P.ABG	0542	
01	A9	68	01	FB	001C8	CALLS	#1, CLISPRESENT		
		02	50	FO	001CB	INSV	R0, #2, #1, INSSGL_CTLMSK+1		
			FF68	CA	9F 001D1	PUSHAB	GLOBAL DSC	0543	
01	A9	68	01	FB	001D5	CALLS	#1, CLISPRESENT		
		04	50	FO	001D8	INSV	R0, #4, #1, INSSGL_CTLMSK+1		
			44	AA	9F 001DE	PUSHAB	P.ABI	0544	
69	01	68	01	FB	001E1	CALLS	#1, CLISPRESENT		
		02	50	FO	001E4	INSV	R0, #2, #1, INSSGL_CTLMSK		
			50	AA	9F 001E9	PUSHAB	P.ABK	0545	
69	01	68	01	FB	001EC	CALLS	#1, CLISPRESENT		
		03	50	FO	001EF	INSV	R0, #3, #1, INSSGL_CTLMSK		
			64	AA	9F 001F4	PUSHAB	P.ABM	0546	
02	A9	68	01	FB	001F7	CALLS	#1, CLISPRESENT		
		00	50	FO	001FA	INSV	R0, #0, #1, INSSGL_CTLMSK+2		
			70	AA	9F 00200	PUSHAB	P.ABO	0547	
01	A9	68	01	FB	00203	CALLS	#1, CLISPRESENT		
		05	50	FO	00206	INSV	R0, #5, #1, INSSGL_CTLMSK+1		
			0088	CA	9F 0020C	PUSHAB	P.ABQ	0548	
01	A9	68	01	FB	00210	CALLS	#1, CLISPRESENT		
		06	50	FO	00213	INSV	R0, #6, #1, INSSGL_CTLMSK+1		
			88	AA	9F 00219	PUSHAB	PRIVILEGED DSC	0549	
01	A9	68	01	FB	0021C	CALLS	#1, CLISPRESENT		
		07	50	FO	0021F	INSV	R0, #7, #1, INSSGL_CTLMSK+1		
			0098	CA	9F 00225	PUSHAB	P.ABS	0550	
02	A9	68	01	FB	00229	CALLS	#1, CLISPRESENT		
		01	50	FO	0022C	INSV	R0, #1, #1, INSSGL_CTLMSK+2		
			00AC	CA	9F 00232	PUSHAB	P.ABU	0551	
02	A9	68	01	FB	00236	CALLS	#1, CLISPRESENT		
		02	50	FO	00239	INSV	R0, #2, #1, INSSGL_CTLMSK+2		
			00BC	CA	9F 0023F	PUSHAB	P.ABW	0552	
02	A9	68	01	FB	00243	CALLS	#1, CLISPRESENT		
		03	50	FO	00246	INSV	R0, #3, #1, INSSGL_CTLMSK+2		
			00D0	CA	9F 0024C	PUSHAB	P.ABY	0553	
02	A9	68	01	FB	00250	CALLS	#1, CLISPRESENT		
		05	50	FO	00253	INSV	R0, #5, #1, INSSGL_CTLMSK+2		
		CF	00	FB	00259	CALLS	#0, PROCESS_COMMAND	0558	
		09	50	E8	0025E	BLBS	STATUS, 22\$		
			50	DD	00261	PUSHL	STATUS		
		00	01	FB	00263	CALLS	#1, LIBSSIGNAL		
		57	08	AC	D0 0026A	MOVL	INTERACTIVE, R7	0564	
		03	57	E9	0026E	BLBC	R7, 23\$		
			FDB7	31	00271	BRW	1\$		
		50	01	D0	00274	MOVL	#1, R0	0565	
				04	00277	RET		0566	

INSPRSOLD
V04-000

INSSPARSE_OLD_CMD

L 1
16-Sep-1984 01:47:51
14-Sep-1984 12:35:40

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSPRSOLD.B32;1

Page 14
(3)

; Routine Size: 632 bytes, Routine Base: \$CODE\$ + 0000

; 370 0567 1

process_command

```

372 0568 1 %SBTTL 'process_command';
373 0569 1
374 0570 1 ROUTINE PROCESS_COMMAND =
375 0571 2 BEGIN
376 0572 2 +++
377 0573 2
378 0574 2 FUNCTIONAL DESCRIPTION:
379 0575 2
380 0576 2 Set up the control flags and call the appropriate routines to execute
381 0577 2 the commands.
382 0578 2
383 0579 2 IMPLICIT INPUT:
384 0580 2
385 0581 2 CLISDCL_PARSE has been called with the input command line.
386 0582 2
387 0583 2 OUTPUT:
388 0584 2
389 0585 2 none
390 0586 2
391 0587 2 IMPLICIT OUTPUT:
392 0588 2
393 0589 2 Set up global control flags, set up FAB for open file, store
394 0590 2 any privilege mask specified and record the open channel number.
395 0591 2
396 0592 2 ROUTINE VALUE:
397 0593 2
398 0594 2 success
399 0595 2 error:
400 0596 2 RMS from file open
401 0597 2 PRV$SETPRIV: if there are invalid privileges specified
402 0598 2
403 0599 2 ---
404 0600 2 OWN
405 0601 2 PRIV_DSC : $BBLOCK [DSC$C_S_BLN]
406 0602 2 PRESET([DSC$B_CLASS] = DSC$K_CLASS_D);
407 0603 2 LOCAL
408 0604 2 CMK_ARGLST : VECTOR [1], ! argument list for change mode dispatcher
409 0605 2 STATUS;
410 0606 2
411 0607 2 |
412 0608 2 | Since INSTALL may be installed with CMKRNL privilege,
413 0609 2 | check that user has CMKRNL privilege before allowing
414 0610 2 | any operations other than /EXIT and /HELP operations.
415 0611 2 |
416 0612 2 CMK_ARGLST [0] = 0;
417 0613 2 STATUS = $CMKRNL (ROUTIN = INSS$CHECK_PRIV, ARGLST = CMK_ARGLST);
418 0614 2 IF NOT .STATUS
419 0615 2 THEN
420 0616 2 BEGIN
421 0617 2 SIGNAL (SS$NOCMKRNL);
422 0618 2 RETURN TRUE;
423 0619 2 END;
424 0620 2
425 0621 2 IF .INSS$GL_CTLMSK [INSS$V_PRIV]
426 0622 2 THEN
427 0623 2 BEGIN
428 0624 2 LOCAL
```



```
.. 429      0625      3      ALL_PRIVS; ! boolean
.. 430      0626      3
.. 431      0627      3      ALL PRIVS = TRUE;                                ! turn on all privs by default
.. 432      0628      3      CH$FILL (0, 8, INSS$GQ_KFEPRIVS);                ! initialize privilege quadword to z
.. 433      0629      3      WHILE CL$GET_VALUE ("PRIVILEGED_DSC, PRIV_DSC ) DO
.. 434      0630      4      BEGIN
.. 435      0631      4      ALL PRIVS = FALSE;                                ! turn off default
.. 436      0632      4      SELECT PRV$SETPRIV (PRIV_DSC, INSS$GQ_KFEPRIVS) OF ! set the appropriate bit in the mask
.. 437      0633      4      SET
.. 438      0634      4
.. 439      0635      4      [PRV$ INVNAM]:
.. 440      0636      5      BEGIN
.. 441      0637      5      SIGNAL (INSS$_INVPRVNAM, 1, PRIV_DSC);
.. 442      0638      5      RETURN TRUE;
.. 443      0639      4      END;
.. 444      0640      4
.. 445      0641      4      [PRV$ NOTUNQ]:
.. 446      0642      5      BEGIN
.. 447      0643      5      SIGNAL (INSS$_PRVNOTUNQ, 1, PRIV_DSC);
.. 448      0644      5      RETURN TRUE;
.. 449      0645      4      END;
.. 450      0646      4      TES;
.. 451      0647      3      END;                                ! end while there are more privs
.. 452      0648      3      IF .ALL PRIVS                                ! No privs specified with /PRIV so
.. 453      0649      3      THEN CH$FILL (-1, 8, INSS$GQ_KFEPRIVS);        ! use default of all privs set
.. 454      0650      3      END;
.. 455      0651      2
.. 456      0652      2      !+++
.. 457      0653      2      :
.. 458      0654      2      Act on the command entered
.. 459      0655      2      :
.. 460      0656      2      !---
.. 461      0657      2
.. 462      0658      3      IF (NOT .INSS$GL_CTLMSK [INSS$V_FILSPC] )
.. 463      0659      2      THEN
.. 464      0660      2      :
.. 465      0661      2      A file spec wasn't entered, therefore the only legal qualifiers
.. 466      0662      2      are /LIST and /GLOBAL and /FULL
.. 467      0663      2      :
.. 468      0664      3      BEGIN
.. 469      0665      3      LOCAL
.. 470      0666      3      STATUS;
.. 471      0667      3
.. 472      0668      4      IF (.INSS$GL_CTLMSK [INSS$V_LIST] OR
.. 473      0669      4      .INSS$GL_CTLMSK [INSS$V_FULL])
.. 474      0670      3      THEN
.. 475      0671      4      BEGIN
.. 476      0672      4      STATUS = INSS$LIST (0);
.. 477      0673      4      IF NOT .STATUS
.. 478      0674      4      THEN
.. 479      0675      4      SIGNAL (((INSS$_FAILED AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
.. 480      0676      4      1, LIST_DSC, .STATUS);
.. 481      0677      4      RETURN TRUE;
.. 482      0678      3      END;
.. 483      0679      3
.. 484      0680      3      IF .INSS$GL_CTLMSK [INSS$V_GLOBAL]
.. 485      0681      3      THEN
```



```
process_command
: 486 0682 4 BEGIN
: 487 0683 4 STATUS = INSS$GLOBAL ();
: 488 0684 4 IF NOT .STATUS
: 489 0685 4 THEN
: 490 0686 4 SIGNAL (((INSS$_FAILED AND NOT STSSM_SEVERITY) OR STSSK_WARNING),
: 491 0687 4 1, GLOBAL_DSC, .STATUS);
: 492 0688 4 RETURN TRUE;
: 493 0689 4 END;
: 494 0690 4 END
: 495 0691 4
: 496 0692 4
: 497 0693 4
: 498 0694 4 A file name was specified
: 499 0695 4
: 500 0696 4
: 501 0697 4 ELSE
: 502 0698 4 BEGIN
: 503 0699 4 LOCAL
: 504 0700 4 OPEN STATUS,
: 505 0701 4 STATUS;
: 506 0702 4
: 507 0703 4
: 508 0704 4 Get the file spec from the CLI
: 509 0705 4
: 510 0706 4 REPORT( CLISGET VALUE ( FILE_SPEC_DSC, INSS$GQ_FILDSC) );
: 511 0707 4 OPEN STATUS = INSS$OPEN_FILE ();
: 512 0708 4 IF NOT .OPEN_STATUS
: 513 0709 4 THEN
: 514 0710 4 INSS$CLOSE_FILE ();
: 515 0711 4
: 516 0712 4 +++
: 517 0713 4
: 518 0714 4 Process file operations
: 519 0715 4
: 520 0716 4 ---
: 521 0717 4 DISPATCH ();
: 522 0718 4
: 523 0719 4 INSS$CLOSE_FILE ();
: 524 0720 4
: 525 0721 4 END;
: 526 0722 4 RETURN TRUE;
: 527 0723 1 END; ! routine process_command
```

```
.PSECT $OWNS$,NOEXE,2
00# 00008 PRIV_DSC:
02 0000B .BYTE 0[3]
0000C .BYTE 2
.BLKB 4
.EXTRN SYSS$CMKRNL
.PSECT $CODE$,NOWRT,2
OFFC 00000 PROCESS_COMMAND:
```


			5B	00000000G	00	9E	000C2	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0570
			5A	0000'	CF	9E	00009	MOVAB	CLISGET_VALUE, R11	
			59	00000000G	00	9E	0000E	MOVAB	PRIV_DSC, R10	
			58	00000000G	00	9E	00015	MOVAB	INSSGQ_KFEPRIVS, R9	
			57	00000000G	00	9E	0001C	MOVAB	LIBSSIGNAL, R8	
					7E	D4	00023	MOVAB	INSSGL_CTLMSK, R7	
					5E	DD	00025	CLRL	CMK_ARGLST	0612
				00000000G	00	9F	00027	PUSHL	SP	0613
		00000000G	00		02	FB	0002D	PUSHAB	INSSCHECK_PRIV	
			0A		50	E8	00034	CALLS	#2, SYSSCMKRN	
			7E	2804	8F	3C	00037	BLBS	STATUS, 1\$	0614
			68		01	FB	0003C	MOVZWL	#10244, -(SP)	0617
					55	11	0003F	CALLS	#1, LIBSSIGNAL	
				01	A7	95	00041	BRB	5\$	0618
					5C	18	00044	TSTB	INSSGL_CTLMSK+1	0621
			56		01	D0	00046	RGEQ	7\$	
08		00	6E		00	2C	00049	MOVL	#1, ALL_PRIVS	0627
					69		0004E	MOVCS	#0, (SP), #0, #8, INSSGQ_KFEPRIVS	0628
					5A	DD	0004F			0629
				0000'	CF	9F	00051	PUSHL	R10	
			6B		02	FB	00055	PUSHAB	PRIVILEGED_DSC	
			3D		50	E9	00058	CALLS	#2, CLISGET_VALUE	
					56	D4	0005B	BLBC	R0, 6\$	
					59	DD	0005D	CLRL	ALL_PRIVS	0631
					5A	DD	0005F	PUSHL	R9	0632
		00000000G	00		02	FB	00061	PUSHL	R10	
		00000000G	52		50	D0	00068	CALLS	#2, PRVSSETPRIV	
			8F		52	D1	0006B	MOVL	R0, R2	
					0C	12	00072	CMPL	R2, #PRVS_INVNAM	0635
					5A	DD	00074	BNEQ	3\$	
					01	DD	00076	PUSHL	R10	0637
				00000000G	8F	DD	00078	PUSHL	#1	
					13	11	0007E	PUSHL	#INSS_INVPRVNAM	
		00000000G	8F		52	D1	00080	BRB	4\$	
					C6	12	00087	CMPL	R2, #PRVS_NOTUNG	0641
					5A	DD	00089	BNEQ	2\$	
					01	DD	0008B	PUSHL	R10	0643
				00000000G	8F	DD	0008D	PUSHL	#1	
			68		03	FB	00093	PUSHL	#INSS_PRVNOTUNG	
					51	11	00096	CALLS	#3, LIBSSIGNAL	
			07		56	E9	00098	BRB	11\$	0644
			6E		00	2C	0009B	BLBC	ALL_PRIVS, 7\$	0648
08	FF	8F			69		000A1	MOVCS	#0, -(SP), #-1, #8, INSSGQ_KFEPRIVS	0649
					67	E8	000A2			0658
			46		01	E0	000A5	BLBS	INSSGL_CTLMSK, 12\$	
		05	01	A7	02	E1	000AA	BBS	#1, INSSGL_CTLMSK+1, 8\$	0668
		17	01	A7	7E	D4	000AF	BBC	#2, INSSGL_CTLMSK+1, 9\$	0669
					01	FB	000B1	CLRL	-(SP)	0672
		00000000G	00		50	D0	000B8	CALLS	#1, INSSLIST	
			52		52	E8	000BB	MOVL	R0, STATUS	
			5F		52	DD	000BE	BLBS	STATUS, 15\$	0673
					CF	9F	000C0	PUSHL	STATUS	0676
				0000'	18	11	000C4	PUSHAB	LIST_DSC	0675
					04	E1	000C6	BRB	10\$	
		52	01	A7	00	FB	000CB	BBC	#4, INSSGL_CTLMSK+1, 15\$	0680
		00000000G	00		50	D0	000D2	CALLS	#0, INSSGLOBAL	0683
			52					MOVL	R0, STATUS	

INSPRSOLD
V04-000

process_command

D 2
16-Sep-1984 01:47:51
14-Sep-1984 12:35:40

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSPRSOLD.B32;1

Page 19
(4)

45		52	E8	000D5	BLBS	STATUS, 15\$	:	0684
		52	DD	000D8	PUSHL	STATUS	:	0687
	0000'	CF	9F	000DA	PUSHAB	GLOBAL_DSC	:	0686
		01	DD	000DE	PUSHL	#1	:	
	00000000*	8F	DD	000E0	PUSHL	#<INSS FAILED8-8>	:	
68		04	FB	000E6	CALLS	#4, LIBSSIGNAL	:	
		32	11	000E9	BRB	15\$	:	0688
	00000000G	00	9F	000EB	PUSHAB	INSSGQ FILDSC	:	0706
	0000'	CF	9F	000F1	PUSHAB	FILE_SPEC_DSC	:	
68		02	FB	000F5	CALLS	#2, CLISGET_VALUE	:	
05		50	E8	000F8	BLBS	STATUS, 13\$	:	
		50	DD	000FB	PUSHL	STATUS	:	
	00000000G	01	FB	000FD	CALLS	#1, LIBSSIGNAL	:	
		00	FB	00100	CALLS	#0, INSSOPEN_FILE	:	0707
	07	50	E8	00107	BLBS	OPEN STATUS, 14\$	:	0708
	00000000G	00	FB	0010A	CALLS	#0, INSSCLOSE_FILE	:	0710
	0000V	CF	00	FB	00111	CALLS	:	0717
	00000000G	00	FB	00116	CALLS	#0, INSSCLOSE_FILE	:	0719
		50	01	D0	0011D	MOVL	:	0722
			04	00120	RET	#1, R0	:	0723

; Routine Size: 289 bytes, Routine Base: \$CODE\$ + 0278

; 528 0724 1

dispatch

```
530 0725 1 %SBTTL 'dispatch';
531 0726 1 ROUTINE DISPATCH =
532 0727 1 |+++
533 0728 1 |
534 0729 1 |---
535 0730 1 |
536 0731 2 BEGIN
537 0732 2 LOCAL
538 0733 2 STATUS;
539 0734 2
540 0735 2 IF NOT .INSSG_KFEFAB[FAB$L_STS]
541 0736 2 AND NOT .INSSGL_CTLMSK[INSSV_DELETE]
542 0737 2 THEN RETURN TRUE;
543 0738 2 IF .INSSGL_CTLMSK [INSSV_DELETE]
544 0739 2 THEN
545 0740 2 |
546 0741 2 | if /delete requested then /replace/create are invalid
547 0742 2 | if /list requested then all the remaining known files will
548 0743 2 | be listed after the delete
549 0744 2 |
550 0745 3 BEGIN
551 0746 3 STATUS = INSSDELETE (); ! delete it
552 0747 3 IF NOT .STATUS
553 0748 3 THEN
554 0749 4 BEGIN
555 0750 4 SIGNAL (((INSS_FAIL AND NOT STSM_SEVERITY) OR STSK_WARNING),
556 0751 4 2,DELETE_DSC, INSSGQ_KFERNS, .STATUS);
557 0752 4
558 0753 4 IF .STATUS EQL INSS_INTRNLERR
559 0754 4 THEN
560 0755 4 SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
561 0756 4 END
562 0757 3 ELSE IF .STATUS EQL INSS_REMOVED
563 0758 3 THEN SIGNAL(.STATUS,T,INSSGQ_KFERNS);
564 0759 4 IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
565 0760 3 THEN REPORT (INSSLIST (0));
566 0761 3 END
567 0762 3
568 0763 3
569 0764 3
570 0765 2 ELSE
571 0766 2 IF .INSSGL_CTLMSK [INSSV_REPLACE]
572 0767 2 THEN
573 0768 2 |
574 0769 2 | Replace has been requested. There can be no other qualifiers.
575 0770 2 | The entry must be deleted and recreated.
576 0771 2 | If /list requested with /replace then do the replace and
577 0772 2 | list the new entry.
578 0773 2 |
579 0774 3 BEGIN
580 0775 3 |
581 0776 3 | Delete and recreate with new attributes
582 0777 3 |
583 0778 3 | INSSGL_CTLMSK [INSSV_DELETE] = TRUE; ! set delete flag
584 0779 3 | INSSGL_REPLACE_MSK = 0;
585 0780 3 | STATUS = INSSDELETE (); ! recreate it
586 0781 3 IF NOT .STATUS
```



```
587 0782 3
588 0783 4
589 0784 6
590 0785 4
591 0786 4
592 0787 4
593 0788 4
594 0789 4
595 0790 4
596 0791 3
597 0792 3
598 0793 3
599 0794 3
600 0795 3
601 0796 3
602 0797 3
603 0798 3
604 0799 3
605 0800 3
606 0801 3
607 0802 3
608 0803 3
609 0804 3
610 0805 3
611 0806 3
612 0807 3
613 0808 4
614 0809 4
615 0810 4
616 0811 4
617 0812 4
618 0813 4
619 0814 4
620 0815 3
621 0816 3
622 0817 4
623 0818 3
624 0819 3
625 0820 3
626 0821 3
627 0822 3
628 0823 3
629 0824 3
630 0825 2
631 0826 2
632 0827 2
633 0828 2
634 0829 2
635 0830 2
636 0831 2
637 0832 2
638 0833 3
639 0834 4
640 0835 3
641 0836 3
642 0837 3
643 0838 4

THEN
  BEGIN
    SIGNAL (((INSS_FAIL AND NOT STSSM_SEVERITY)
              OR STSSK_WARNING),
            2, DELETE_DSC, INSSGQ_KFERNS, .STATUS);

    IF .STATUS EQL INSS_INTRNLERR
    THEN
      SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
    END;
    INSSGL_KFEADR = 0;          ! Entry has gone away
    INSCLOSE_FILE ();

    INSSOPEN_FILE ();
    INSSGL_REPLACE_MSK [INSSV_LIST] = .INSSGL_CTLMSK [INSSV_LIST];
    INSSGL_REPLACE_MSK [INSSV_FULL] = .INSSGL_CTLMSK [INSSV_FULL];
    INSSGL_CTLMSK = .INSSGL_REPLACE_MSK;      ! use replace mask
    INSSGL_CTLMSK [INSSV_CREATE] = TRUE;      ! set create flag
    INSSGL_CTLMSK [INSSV_DELETE] = FALSE;     ! clear delete flag
    IF .INSSGL_KFEADR EQL 0
    THEN
      STATUS = INSSCREATE ()          ! recreate it
    ELSE
      STATUS = INSEXISTS;

    IF NOT .STATUS
    THEN
      BEGIN
        SIGNAL (INSS_FAIL, 2, REPLACE_DSC, INSSGQ_KFERNS, .STATUS);

        IF .STATUS EQL INSS_INTRNLERR
        THEN
          SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
        END;

        IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
        THEN REPORT (INSSLIST (.INSSGL_KFEADR));
        IF .INSSGL_CTLMSK [INSSV_GLOBAL]
        THEN REPORT (INSSGLOBAL (.INSSGL_KFEADR));
      END          ! /REPLACE

    ELSE
      : filespec given but not for replace or delete.
      : A /LIST without /CREATE requests list of given file's KFE.
      : A /LIST with /CREATE requests file be added to KFE and
      : that entry be listed.
      : A filename without /LIST or /CREATE is an implied /CREATE.
      BEGIN
        IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL] OR
            .INSSGL_CTLMSK [INSSV_GLOBAL]) AND
            NOT .INSSGL_CTLMSK [INSSV_CREATE]
        THEN
          BEGIN
```



```
644 0839 4 IF .INSSGL_KFEADR EQL 0
645 0840 4 THEN
646 0841 5 BEGIN
647 0842 7 SIGNAL (((INSS_FAIL AND NOT STS$M_SEVERITY)
648 0843 5 OR STS$K_WARNING),
649 0844 5 2, LIST_DSC, INSSGQ_KFERNS, INSS_NOKFEFND);
650 0845 5 RETURN TRUE;
651 0846 4 END;
652 0847 4 END
653 0848 3 ELSE
654 0849 4 BEGIN
655 0850 4 INSSGL_CTLMSK [INSSV_CREATE] = TRUE; ! set create flag
656 0851 4 IF .INSSGL_KFEADR EQL 0
657 0852 4 THEN
658 0853 4 STATUS = INSS_CREATE () ! Create it
659 0854 4 ELSE
660 0855 4 STATUS = INSS_EXISTS;
661 0856 4
662 0857 4 IF NOT .STATUS
663 0858 4 THEN
664 0859 5 BEGIN
665 0860 5 SIGNAL (INSS_FAIL, 2, CREATE_DSC, INSSGQ_KFERNS, .STATUS);
666 0861 5
667 0862 5 IF .STATUS EQL INSS_INTRNLERR
668 0863 5 THEN
669 0864 5 SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
670 0865 4 END;
671 0866 3 END;
672 0867 3
673 0868 4 IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
674 0869 3 THEN REPORT (INSSLIST (.INSSGL_KFEADR));
675 0870 3 IF .INSSGL_CTLMSK [INSSV_GLOBAL]
676 0871 3 THEN REPORT (INSSGLOBAL (.INSSGL_KFEADR));
677 0872 2 END;
678 0873 2 RETURN TRUE;
679 0874 1 END; ! Routine dispatch
```

OFFC 00000 DISPATCH:

5B	0000'	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0726
5A	00000000G	00	9E	00007	MOVAB	DELETE_DSC, R11	:
59	00000000G	00	9E	0000E	MOVAB	INSSLIST, R10	:
58	00000000G	00	9E	00015	MOVAB	INSSGL_REPLACE_MSK, R9	:
57	00000000G	00	9E	0001C	MOVAB	INSSL_INTRNLERR, R8	:
56	00000000G	00	9E	00023	MOVAB	INSSGL_KFEADR, R7	:
55	00000000G	8F	D0	0002A	MOVL	INSSGQ_KFERNS, R6	:
54	00000000G	00	9E	00031	MOVAB	#INSS_INTRNLERR, R5	:
53	00000000G	00	9E	00038	MOVAB	LIB\$SIGNAL, R4	:
07	00000000G	00	E8	0003F	BLBS	INSSGL_CTLMSK, R3	:
63		06	E0	00046	BBS	INSSG_RFEFAB+8, 2\$	: 0735
		01AA	31	0004A 1\$:	BRW	#6, INSSGL_CTLMSK, 3\$	: 0736
4F		06	E1	0004D 2\$:	BBC	26\$	: 0737
00000000G	00	00	FB	00051 3\$:	CALLS	#6, INSSGL_CTLMSK, 8\$	: 0738
						#0, INSSDELETE	: 0746

			52		50	DO	00058	MOVL	R0, STATUS		
			1E		52	E8	0005B	BLBS	STATUS, 4\$		0747
					52	DD	0005E	PUSHL	STATUS		0751
					56	DD	00060	PUSHL	R6		0750
					5B	DD	00062	PUSHL	R11		
					02	DD	00064	PUSHL	#2		
				00000000*	8F	DD	00066	PUSHL	#<INSS_FAIL8-8>		
		64			05	FB	0006C	CALLS	#5, LIBSSIGNAL		
		55			52	D1	0006F	CMPL	STATUS, R5		0753
					1A	12	00072	BNEQ	6\$		
					68	DD	00074	PUSHL	INSSL_INTRNLERR		0755
					01	DD	00076	PUSHL	#1		
					55	DD	00078	PUSHL	R5		
					0F	11	0007A	BRB	5\$		
			00000000G	8F	52	D1	0007C	CMPL	STATUS, #INSS_REMOVED		0757
					09	12	00083	BNEQ	6\$		
					56	DD	00085	PUSHL	R6		0758
					01	DD	00087	PUSHL	#1		
					52	DD	00089	PUSHL	STATUS		
		64			03	FB	0008B	CALLS	#3, LIBSSIGNAL		
	05	01	A3		01	E0	0008E	BBS	#1, INSSGL_CTLMSK+1, 7\$		0759
	B2	01	A3		02	E1	00093	BBC	#2, INSSGL_CTLMSK+1, 1\$		
					7E	D4	00098	CLRL	-(SP)		0760
			6A		01	FB	0009A	CALLS	#1, INSSLIST		
					014F	31	0009D	BRW	25\$		
	03		63		05	E0	000A0	BBS	#5, INSSGL_CTLMSK, 9\$		0766
					00B6	31	000A4	BRW	16\$		
			63		40	8F	88	000A7	BISB2	#64, INSSGL_CTLMSK	0778
						69	D4	000AB	CLRL	INSSGL_REPLACE_MSK	0779
			00000000G	00	00	FB	000AD	CALLS	#0, INSSDELETE		0780
				52	50	DO	000B4	MOVL	R0, STATUS		
			1F		52	E8	000B7	BLBS	STATUS, 10\$		0781
					52	DD	000BA	PUSHL	STATUS		0786
					56	DD	000BC	PUSHL	R6		0784
					5B	DD	000BE	PUSHL	R11		
					02	DD	000C0	PUSHL	#2		
				00000000*	8F	DD	000C2	PUSHL	#<INSS_FAIL8-8>		
		64			05	FB	000C8	CALLS	#5, LIBSSIGNAL		
		55			52	D1	000CB	CMPL	STATUS, R5		0788
					09	12	000CE	BNEQ	10\$		
					68	DD	000D0	PUSHL	INSSL_INTRNLERR		0790
					01	DD	000D2	PUSHL	#1		
					55	DD	000D4	PUSHL	R5		
			64		03	FB	000D6	CALLS	#3, LIBSSIGNAL		
					67	D4	000D9	CLRL	INSSGL_KFEADR		0792
			00000000G	00	00	FB	000DB	CALLS	#0, INSSCLOSE FILE		0793
			00000000G	00	00	FB	000E2	CALLS	#0, INSSOPEN FILE		0795
				01	01	EF	000E9	EXTZV	#1, #1, INSSGL_CTLMSK+1, R0		0796
01	50	01	A3		50	F0	000EF	INSV	R0, #1, #1, INSSGL_REPLACE_MSK+1		
	A9		01		02	EF	000F5	EXTZV	#2, #1, INSSGL_CTLMSK+1, R0		0797
					50	F0	000FB	INSV	R0, #2, #1, INSSGL_REPLACE_MSK+1		
01	50		A3		69	DO	00101	MOVL	INSSGL_REPLACE_MSK, INSSGL_CTLMSK		0798
	A9		01		63						
					63	10	88	00104	BISB2	#16, INSSGL_CTLMSK	0799
					63	40	8F	8A	00107	BICB2	#64, INSSGL_CTLMSK
							67	D5	0010B	TSTL	INSSGL_KFEADR
							0C	12	0010D	BNEQ	11\$
			00000000G	00	00	FB	0010F	CALLS	#0, INSSCREATE		0803

		52		50	DO 00116	MOVL	R0, STATUS		
				07	11 00119	BRB	12\$		
		52	00000000G	8F	DO 0011B 11\$:	MOVL	#INSS EXISTS, STATUS		0805
		20		52	E8 00122 12\$:	BLBS	STATUS, 13\$		0807
				52	DD 00125	PUSHL	STATUS		0810
				56	DD 00127	PUSHL	R6		
			FO	AB	9F 00129	PUSHAB	REPLACE_DSC		
				02	DD 0012C	PUSHL	#2		
			00000000G	8F	DD 0012E	PUSHL	#INSS FAIL		
		64		05	FB 00134	CALLS	#5, LIBSSIGNAL		
		55		52	D1 00137	CMPL	STATUS, R5		0812
				09	12 0013A	BNEQ	13\$		
				68	DD 0013C	PUSHL	INSSL_INTRNLERR		0814
				01	DD 0013E	PUSHL	#1		
				55	DD 00140	PUSHL	R5		
		64		03	FB 00142	CALLS	#3, LIBSSIGNAL		
05	01	A3		01	E0 00145 13\$:	BBS	#1, INSSGL_CTLMSK+1, 14\$		0817
0B	01	A3		02	E1 0014A	BBC	#2, INSSGL_CTLMSK+1, 15\$		
				67	DD 0014F 14\$:	PUSHL	INSSGL_KFEADR		0818
		6A		01	FB 00151	CALLS	#1, INSSLIST		
		03		50	E8 00154	BLBS	STATUS, 15\$		
				0082	31 00157	BRW	23\$		
				0084	31 0015A 15\$:	BRW	24\$		
		50		67	DO 0015D 16\$:	MOVL	INSSGL_KFEADR, R0		0839
0A	01	A3		01	E0 00160	BBS	#1, INSSGL_CTLMSK+1, 17\$		0834
05	01	A3		02	E0 00165	BBS	#2, INSSGL_CTLMSK+1, 17\$		
1E	01	A3		04	E1 0016A	BBC	#4, INSSGL_CTLMSK+1, 18\$		0835
1A		63		04	E0 0016F 17\$:	BBS	#4, INSSGL_CTLMSK, 18\$		0836
				55	12 00173	BNEQ	21\$		0839
			00000000G	8F	DD 00175	PUSHL	#INSS_NOKFEFND		0842
				56	DD 0017B	PUSHL	R6		
			BC	AB	9F 0017D	PUSHAB	LIST_DSC		
				02	DD 00180	PUSHL	#2		
			00000000*	8F	DD 00182	PUSHL	#<INSS_FAIL&-8>		
		64		05	FB 00188	CALLS	#5, LIBSSIGNAL		
				6A	11 0018B	BRB	26\$		0845
		63		10	88 0018D 18\$:	BISB2	#16, INSSGL_CTLMSK		0850
				50	D5 00190	TSTL	R0		0851
				0C	12 00192	BNEQ	19\$		
		00	00000000G	00	FB 00194	CALLS	#0, INSSCREATE		0853
		52		50	DO 0019B	MOVL	R0, STATUS		
				07	11 0019E	BRB	20\$		
		52	00000000G	8F	DO 001A0 19\$:	MOVL	#INSS EXISTS, STATUS		0855
		20		52	E8 001A7 20\$:	BLBS	STATUS, 21\$		0857
				52	DD 001AA	PUSHL	STATUS		0860
				56	DD 001AC	PUSHL	R6		
			EO	AB	9F 001AE	PUSHAB	CREATE_DSC		
				02	DD 001B1	PUSHL	#2		
			00000000G	8F	DD 001B3	PUSHL	#INSS FAIL		
		64		05	FB 001B9	CALLS	#5, LIBSSIGNAL		
		55		52	D1 001BC	CMPL	STATUS, R5		0862
				09	12 001BF	BNEQ	21\$		
				68	DD 001C1	PUSHL	INSSL_INTRNLERR		0864
				01	DD 001C3	PUSHL	#1		
				55	DD 001C5	PUSHL	R5		
		64		03	FB 001C7	CALLS	#3, LIBSSIGNAL		
05	01	A3		01	E0 001CA 21\$:	BBS	#1, INSSGL_CTLMSK+1, 22\$		0868

dispatch

J 2
16-Sep-1984 01:47:51
14-Sep-1984 12:35:40

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSPRSOLD.B32;1

Page 25
(5)

0D 01 A3

6A
0564
A3

11

01

00000000G

00
0564
50

02	E1	001CF	
67	DD	001D4	22\$:
01	FB	001D6	
50	E8	001D9	
50	DD	001DC	23\$:
01	FB	001DE	
04	E1	001E1	24\$:
67	DD	001E6	
01	FB	001E8	
50	E8	001EF	25\$:
50	DD	001F2	
01	FB	001F4	
01	DD	001F7	26\$:
	04	001FA	

```

BBC      #2, IN$SGL CTLMSK+1, 24$
PUSHL   IN$SGL KFEADR
CALLS   #1, IN$LIST
BLBS    STATUS, 24$
PUSHL   STATUS
CALLS   #1, LIB$SIGNAL
BBC      #4, IN$SGL CTLMSK+1, 26$
PUSHL   IN$SGL KFEADR
CALLS   #1, IN$GLOBAL
BLBS    STATUS, 26$
PUSHL   STATUS
CALLS   #1, LIB$SIGNAL
MOVL    #1, RO
RET

```

0869
0870
0871
0873
0874

; Routine Size: 507 bytes, Routine Base: \$CODE\$ + 0399

; 680 0875 1

help

```

: 682 0876 1 %SBTTL 'help';
: 683 0877 1
: 684 0878 1 ROUTINE HELP ( HLP_DSC) =
: 685 0879 1 +++
: 686 0880 1
: 687 0881 1 FUNCTIONAL DESCRIPTION:
: 688 0882 1 Conduct interactive help session
: 689 0883 1
: 690 0884 1 INPUT:
: 691 0885 1 hlp_dsc : Descriptor of command, possibly with qualifier(s) for
: 692 0886 1 which help is requested
: 693 0887 1
: 694 0888 1 OUTPUT:
: 695 0889 1 Interactive help session
: 696 0890 1
: 697 0891 1 ROUTINE VALUE:
: 698 0892 1 Low bit set for success
: 699 0893 1 unless an error is returned from lbr$output_help.
: 700 0894 1 ---
: 701 0895 2 BEGIN
: 702 0896 2 LOCAL
: 703 0897 2 HLP_FLAGS;
: 704 0898 2
: 705 0899 2 HLP_FLAGS = HLP$M_PROCESS OR HLP$M_GROUP OR HLP$M_SYSTEM;
: 706 P 0900 2 REPORT ( LBR$OUTPUT_HELP (LIB$PUT_OUTPUT, 0, .HLP_DSC,
: 707 0901 2 $DESCRIPTOR('INSTALHLP'), HLP_FLAGS, LIB$GET_INPUT) );
: 708 0902 2 RETURN TRUE;
: 709 0903 1 END;
```

.PSECT \$PLITS\$,NOWRT,NOEXE,2

```

50 4C 48 4C 41 54 53 4E 49 00198 P.ACB: .ASCII \INSTALHLP\
                                001A1 .BLKB 3
                                00000009 001A4 P.ACA: .LONG 9
                                00000000 001A8 .ADDRESS P.ACB
```

.PSECT \$CODE\$,NOWRT,2

```

                                0000 00000 HELP: .WORD Save nothing
                                0E DD 00002 PUSHL #14
                                00000000G 00 9F 00004 PUSHAB LIB$GET_INPUT
                                04 AE 9F 0000A PUSHAB HLP_FLAGS
                                0000' CF 9F 0000D PUSHAB P.ACA
                                04 AC DD 00011 PUSHL HLP_DSC
                                7E D4 00014 CLRL -(SP)
                                00000000G 00 9F 00016 PUSHAB LIB$PUT_OUTPUT
                                00000000G 00 06 FB 0001C CALLS #6, LBR$OUTPUT_HELP
                                00 50 E8 00023 BLBS STATUS, 1$
                                00 50 DD 00026 PUSHL STATUS
                                00000000G 00 01 FB 00028 CALLS #1, LIB$SIGNAL
                                50 01 D0 0002F 1$: MOVL #1, R0
                                04 00032 RET
```

```

: 0878
: 0899
: 0901
:
:
:
:
:
:
:
:
: 0902
: 0903
```


INSPRSOLD
V04-000 help

L 2
16-Sep-1984 01:47:51
14-Sep-1984 12:35:40

VAX-11 Bliss-32 V4.0-742
[INSTAL.SRC]INSPRSOLD.B32;1

Page 27
(6)

; Routine Size: 51 bytes, Routine Base: \$CODE\$ + 0594

: 710 0904 1
: 711 0905 1 END ! Module INSPRSOLD
: 712 0906 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	16	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	428	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	1479	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	22	0	1000	00:01.9

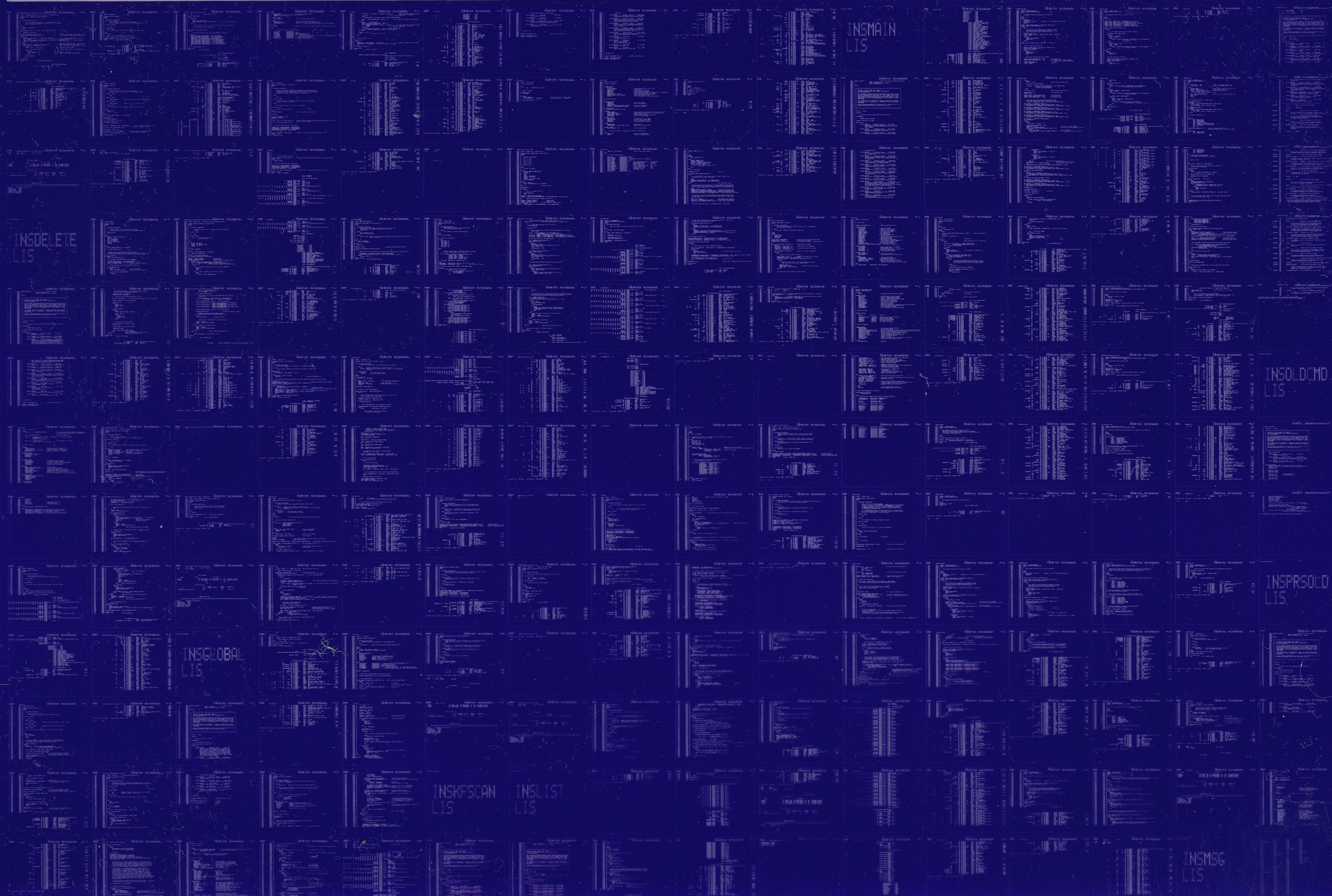
COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:INSPRSOLD/OBJ=OBJ\$:INSPRSOLD MSRC\$:INSPRSOLD/UPDATE=(ENH\$:INSPRSOLD)

; Size: 1479 code + 444 data bytes
; Run Time: 00:25.3
; Elapsed Time: 01:40.8
; Lines/CPU Min: 2146
; Lexemes/CPU-Min: 15259
; Memory Used: 244 pages
; Compilation Complete

0189 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



INSKFSAN
LIS

INSLIST
LIS

INMSG
LIS

0190 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

