



```

IIIIII  NN  NN  SSSSSSS  PPPPPPP  RRRRRRR  SSSSSSS  00000  LL  DDDDDDD
IIIIII  NN  NN  SSSSSSS  PPPPPPP  RRRRRRR  SSSSSSS  00000  LL  DDDDDDD
  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  PPPPPPP  RRRRRRR  SSSSSS  00        00  LL  DD        DD
  II    NN  NN  SSSSSS  PPPPPPP  RRRRRRR  SSSSSS  00        00  LL  DD        DD
  II    NN  NN  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  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  SSSSSSS  PPPPPPP  RRRRRRR  SSSSSSS  00000  LL  DDDDDDD
IIIIII  NN  NN  SSSSSSS  PPPPPPP  RRRRRRR  SSSSSSS  00000  LL  DDDDDDD

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
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  SSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSS

```

```

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.

```

: 58  
: 59  
: 60  
: 61  
: 62  
: 63  
: 64  
: 65  
: 66  
: 67  
: 68  
: 69

0058 1  
0059 1  
0060 1  
0061 1  
0062 1  
0063 1  
0064 1  
0065 1  
0066 1  
0067 1  
0068 1  
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 'SYSS$LIBRARY:LIB'; | VAX/VMS system definitions
: 78 0077 1
: 79 0078 1 REQUIRE 'SRC$:INSPREFIX.REQ';
: 80 0220 1 REQUIRE 'LIB$:INSDEF.R32';
: 81 0279 1
: 82 0280 1
: 83 0281 1 |
: 84 0282 1 | Table of contents
: 85 0283 1 |
: 86 0284 1
: 87 0285 1 FORWARD ROUTINE
: 88 0286 1 PROCESS COMMAND, | execute the command line
: 89 0287 1 DISPATCH, | check for conflicting quals and
: 90 0288 1 | dispatch to appropriate routine
: 91 0289 1 HELP; | print interactive help
: 92 0290 1
: 93 0291 1 EXTERNAL
: 94 0292 1 INSOLDCMD; | Tables describing the old Install commands
: 95 0293 1
: 96 0294 1 |
: 97 0295 1 | External routines
: 98 0296 1 |
: 99 0297 1
100 0298 1 EXTERNAL ROUTINE
101 0299 1 ! Module INSMAN
102 0300 1 !NSSOPEN_FILE, | handle file open processing
103 0301 1 !NSSCLOSE_FILE, | handle file close
104 0302 1 !NSSCHECK_PRIV, | enforce privilege check in case INSTALL is installed
105 0303 1
106 0304 1 !NSSCREATE, | perform CREATE command
107 0305 1 !NSSDELETE, | perform DELETE command
108 0306 1 !NSSGLOBAL, | list global sections
109 0307 1 !NSSLIST; | list known file entries
110 0308 1
111 0309 1 EXTERNAL ROUTINE
112 0310 1 CLISDCL_PARSE, | Parse DCL command
113 0311 1 CLISPRESENT, | Check if entity present
114 0312 1 CLISGET_VALLE, | Get value from command line
115 0313 1 LBR$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 PRV$SETPRIV, | set privilege bits given ASCII string
119 0317 1 STR$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

```

Declarations

```

128 0326 1 PRVS_INVNAM, ! Invalid privilege name
129 0327 1 PRVS_NOTUNQ; ! Privilege name not unique
130 0328 1
131 0329 1 EXTERNAL LITERAL
132 0330 1 INSS_EXISTS, ! Known File Entry already exists
133 0331 1 INSS_FAIL, ! Unable to <OPERATION> <FILENAME>
134 0332 1 INSS_FAILED, ! Failed to process request for <FILENAME>
135 0333 1 INSS_HELP, ! Enter /HELP for INSTALL help
136 0334 1 INSS_INTRNLERR, ! Internal Error: error detail
137 0335 1 INSS_INVPRVAM, ! Invalid privilege name
138 0336 1 INSS_NOKFEFND, ! Known File not found
139 0337 1 INSS_NOLOAD, ! Network and sequentially loaded files are not installable
140 0338 1 INSS_PRVNOTUNQ, ! Privilege name not unique
141 0339 1 INSS_REMOVED; ! Known entry was removed
142 0340 1
143 0341 1 EXTERNAL
144 0342 1 INSSL_INTRNLERR, ! Store internal error descriptor
145 0343 1 INSSG_KFENAM, ! file name block declaration
146 0344 1 INSSG_KFECHAN, ! channel for open known file
147 0345 1 INSSGL_CTLMSK : $BBLOCK, ! Control flags
148 0346 1 INSSGL_REPLACE_MSK: $BBLOCK, ! Control flags saved for REPLACE
149 0347 1 INSSG_KFERNS : ! buffer for resultant name
150 0348 1 $BBLOCK,
151 0349 1 INSSGL_KFEADR, ! id of file to be listed
152 0350 1 INSSG_KFEPRIVS : $BBLOCK, ! Quadword privilege mask
153 0351 1 INSSG_FILDSC : ! descriptor to get filename returned
154 0352 1 $BBLOCK, ! from CLI.
155 0353 1 INSSG_KFEFAB : $BBLOCK, ! file access block declaration
156 0354 1
157 0355 1 CTL$GQ_PROCPRIV : $BBLOCK; ! privilege mask of user
158 0356 1
159 0357 1 OWN
160 0358 1 INPUT_DESC : DYN_DESC_DECL; ! command string input
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_HELP_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

```

```
INS$PARSE_OLD_CMD
177 0374 1 %SBTTL 'INS$PARSE_OLD_CMD';
178 0375 1
179 0376 1 GLOBAL ROUTINE INS$PARSE_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, $DESCRIPTOR('INSTALL>') );
209 0406 2 IF .STATUS EQL RMS$_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 ); ! Upcase 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;
286 0483 4 IF .HLP_PTR EQL 0 ! If not found, look for "/H
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
296 0493 4      BEGIN
297 0494 5
298 0495 5          replace "/HEL" or "/HELP" with "INSTALL" since lbr$output_help
299 0496 5          needs to see the topic keyword.
300 0497 5          Then call the help routine.
301 0498 5
302 0499 5      STR$RIGHT ( .PARSE_CMD_DESC, .PARSE_CMD_DESC, %REF (.HLP_PTR+.HLP_LEN) );
303 0500 5      STR$CONCAT (.PARSE_CMD_DESC, $DESCRIPTOR ('INSTALL '), .PARSE_CMD_DESC);
304 0501 5          ! Stick "INSTALL" on front of help line
305 0502 5      REPORT( HELP ( .PARSE_CMD_DESC ));
306 0503 5      END
307 0504 5          ! Help was requested
308 0505 5
309 0506 4      ELSE
310 0507 5      BEGIN
311 0508 5          ! Command was something other than /HELP
312 0509 5
313 0510 5      !+++
314 0511 5          No help requested, proceed to parsing command line
315 0512 5          Must concatenate command line to "INSTALL" string for
316 0513 5          cli$dcl_parse to execute.
317 0514 5
318 0515 5      ---
319 0516 5      STR$CONCAT (.PARSE_CMD_DESC, $DESCRIPTOR ('INSTALL '), .PARSE_CMD_DESC); ! Stick "INSTALL" on front
320 0517 5      STATUS = CLISDCL_PARSE (.PARSE_CMD_DESC, INSOLDCMD ); ! parse the command line
321 0518 5
322 0519 5      IF .STATUS
323 0520 5      THEN
324 0521 6      BEGIN
325 0522 6          IF CLISPRESENT( EXIT_DSC ) ! If /EXIT, then exit program
326 0523 6          THEN EXITLOOP;
327 0524 6
328 0525 6      !+++
329 0526 6          Set flags for the qualifiers present in command line
330 0527 6
331 0528 6      !---
332 0529 6
333 0530 6      INSSGL_CTLMSK = 0; ! clear flags
334 0531 6      INSSL_INTRNLERR = 0; ! clear internal error descriptor
335 0532 6
336 0533 6      INSTALL control flags
337 0534 6
338 0535 6      INSSGL_CTLMSK [INSSV_FILSPC] = CLISPRESENT( FILE_SPEC_DSC );
339 0536 6
340 0537 6      INSSGL_CTLMSK [INSSV_CREATE] = CLISPRESENT( CREATE_DSC );
341 0538 6      INSSGL_CTLMSK [INSSV_DELETE] = CLISPRESENT( DELETE_DSC );
342 0539 6      INSSGL_CTLMSK [INSSV_REPLACE] = CLISPRESENT( REPLACE_DSC );
343 0540 6      INSSGL_CTLMSK [INSSV_LIST] = CLISPRESENT( LIST_DSC );
344 0541 6      INSSGL_CTLMSK [INSSV_FULL] = CLISPRESENT( %ASCID 'FULL' );
345 0542 6      INSSGL_CTLMSK [INSSV_GLOBAL] = CLISPRESENT( GLOBAL_DSC );
346 0543 6      INSSGL_CTLMSK [INSSV_HELP] = CLISPRESENT( %ASCID 'HELP' );
347 0544 6

```





```

59 4C 4E 4F 5F 45 54 55 43 45 00000000' 00180
58 45 00184 P.ABZ:
010E000C 00190 P.ABY:
00000000' 00194

```

```

.ADDRESS P.ABX
.ASCII \EXECUTE_ONLY\
.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_INVPRVNAM
.EXTRN INSS_NOKFEFND, INSS_NOLOAD
.EXTRN INSS_PRVNOTUNQ, INSS_REMOVED
.EXTRN INSS_INTRNLERR
.EXTRN INSSG_KFENAM, INSSGL_KFECHAN
.EXTRN INSSG_CTLMSK, INSSG_REPLACE_MSK
.EXTRN INSSG_KFERNS, INSSGL_KFEADR
.EXTRN INSSG_KFEPRIVS
.EXTRN INSSG_FILDSC, INSSG_KFEFAB
.EXTRN CTL$GQ_PROCPRIV

```

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

```

.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

```

```

5B 0000' CF 9E 00002
5A 0000' CF 9E 00007
59 0000000G 00 9E 0000C
58 0000000G 00 9E 00013

```

		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	1\$:	PUSHL	R10		0405
				5B	DD	0002D		PUSHL	R11		
	00000000G	00		02	FB	0002F		CALLS	#2, LIB\$GET_INPUT		
		55		50	D0	00036		MOVL	R0, STATUS		
	0001827A	8F		55	D1	00039		CML	STATUS, #98938		0406
				03	12	00040		BNEQ	2\$		
				022F	31	00042		BRW	23\$		
		0D		55	E8	00045	2\$:	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	3\$:	MOVL	INPUT_DESC+4, R3		0416
	63	6B		2D	3A	00059		LOCC	#45, INPUT_DESC, (R3)		0415
				02	12	0005D		BNEQ	4\$		
				51	D4	0005F		CLRL	R1		
		54		51	D0	00061	4\$:	MOVL	R1, DASH_PTR		
		52	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
		07	AE	01	90	0006F		MOVW	#1, COPY_DESC+3		0421
	04	AE		53	A3	00073		SUBW3	R3, DASH_PTR, COPY_DESC		0422
		08	AE	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	5\$:	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		CML	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	6\$:	MOVL	INPUT_DESC+4, R3		0439
		08	AE	53	D0	000AC		MOVL	R3, COPY_DESC+4		
	63	6B		2D	3A	000B0		LOCC	#45, INPUT_DESC, (R3)		0441
				02	12	000B4		BNEQ	7\$		
				51	D4	000B6		CLRL	R1		
		54		51	D0	000B8	7\$:	MOVL	R1, DASH_PTR		
				08	12	000BB		BNEQ	8\$		0442
				56	D4	000BD		CLRL	LAST_WAS_CONT		0445
		04	AE	6B	B0	000BF		MOVW	INPUT_DESC, COPY_DESC		0446
				05	11	000C3		BRB	9\$		0442
	04	AE		53	A3	000C5	8\$:	SUBW3	R3, DASH_PTR, COPY_DESC		0450
		54	04	AE	9F	000CA	9\$:	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\$	
		52	DD	000E1	11\$:		PUSHL	R2	: 0456
		02	11	000E3			BRB	13\$	
		5B	DD	000E5	12\$:		PUSHL	R11	: 0459
	00000000G	00	52	DD	000E7	13\$:	PUSHL	R2	
			02	FB	000E9		CALLS	#2, STR\$UPCASE	
			62	B5	000F0	14\$:	TSTW	(R2)	: 0468
			7F	13	000F2		BEQL	18\$	
04	B2	62	20	3B	000F4		SKPC	#32, (R2), @4(R2)	: 0470
			02	12	000F9		BNEQ	15\$	
			51	D4	000FB		CLRL	R1	
			51	D5	000FD	15\$:	TSTL	R1	: 0471
			72	13	000FF		BEQL	18\$	
			04	AA	9F	00101	PUSHAB	SLASH_HELP_DSC	: 0481
			52	DD	00104		PUSHL	R2	
	00000000G	00	02	FB	00106		CALLS	#2, STR\$POSITION	
		53	05	D0	0010D		MOVL	#5, HLP_LEN	: 0482
			50	D5	00110		TSTL	HLP_PTR	: 0483
			0F	12	00112		BNEQ	16\$	
			C4	AA	9F	00114	PUSHAB	SLASH_HEL_DSC	: 0486
			52	DD	00117		PUSHL	R2	
	00000000G	00	02	FB	00119		CALLS	#2, STR\$POSITION	
		53	04	D0	00120		MOVL	#4, HLP_LEN	: 0487
			50	D5	00123	16\$:	TSTL	HLP_PTR	: 0489
			29	13	00125		BEQL	17\$	
6E		50	53	C1	00127		ADDL3	HLP_LEN, HLP_PTR, (SP)	: 0500
			4004	8F	BB	0012B	PUSHR	#^MZR2, SP>	
			52	DD	0012F		PUSHL	R2	
	00000000G	00	03	FB	00131		CALLS	#3, STR\$RIGHT	
			52	DD	00138		PUSHL	R2	: 0501
			1C	AA	9F	0013A	PUSHAB	P.ABC	
			52	DD	0013D		PUSHL	R2	
	00000000G	00	03	FB	0013F		CALLS	#3, STR\$CONCAT	
			52	DD	00146		PUSHL	R2	: 0503
	0000V	CF	01	FB	00148		CALLS	#1, HELP	
			010E	31	0014D		BRW	21\$	
			52	DD	00150	17\$:	PUSHL	R2	: 0516
			2C	AA	9F	00152	PUSHAB	P.ABE	
			52	DD	00155		PUSHL	R2	
	00000000G	00	03	FB	00157		CALLS	#3, STR\$CONCAT	
			00000000G	00	9F	0015E	PUSHAB	INSOLD CMD	: 0517
			52	DD	00164		PUSHL	R2	
	00000000G	00	02	FB	00166		CALLS	#2, CLISDCL_PARSE	
		55	50	D0	0016D		MOVL	R0, STATUS	
		03	55	E8	00170		BLBS	STATUS, 19\$	: 0519
			00F4	31	00173	18\$:	BRW	22\$	
			FF44	CA	9F	00176	PUSHAB	EXIT_DSC	: 0522
		68	01	FB	0017A		CALLS	#1, CLISPRESENT	
		03	50	E9	0017D		BLBC	R0, 20\$	
			00F1	31	00180		BRW	23\$	
			69	D4	00183	20\$:	CLRL	INSSGL_CTLMSK	: 0531
			00000000G	00	D4	00185	CLRL	INSSL_INTRNLERR	: 0532
			FF58	CA	9F	0018B	PUSHAB	FILE_SPEC_DSC	: 0536
		68	01	FB	0018F		CALLS	#1, CLISPRESENT	
69			50	F0	00192		INSV	R0, #0, #1, INSSGL_CTLMSK	
	01	00	98	AA	9F	00197	PUSHAB	CREATE_DSC	: 0538

69	01	68 04	01 50	FB FO	0019A 0019D	CALLS INSV	#1, CLISPRESENT R0, #4, #1, INSSGL_CTLMSK	0539
			B8	AA	9F 001A2	PUSHAB	DELETE DSC	
69	01	68 06	01 50	FB FO	001A5 001A8	CALLS INSV	#1, CLISPRESENT R0, #6, #1, INSSGL_CTLMSK	0540
			A8	AA	9F 001AD	PUSHAB	REPLACE DSC	
69	01	68 05	01 50	FB FO	001B0 001B3	CALLS INSV	#1, CLISPRESENT R0, #5, #1, INSSGL_CTLMSK	0541
			FF74	CA	9F 001B8	PUSHAB	LIST DSC	
01	A9	68 01	01 50	FB FO	001BC 001BF	CALLS INSV	#1, CLISPRESENT R0, #1, #1, INSSGL_CTLMSK+1	0542
			38	AA	9F 001C5	PUSHAB	P.ABG	
01	A9	68 02	01 50	FB FO	001C8 001CB	CALLS INSV	#1, CLISPRESENT R0, #2, #1, INSSGL_CTLMSK+1	0543
			FF68	CA	9F 001D1	PUSHAB	GLOBAL DSC	
01	A9	68 04	01 50	FB FO	001D5 001D8	CALLS INSV	#1, CLISPRESENT R0, #4, #1, INSSGL_CTLMSK+1	0544
			44	AA	9F 001DE	PUSHAB	P.ABI	
69	01	68 02	01 50	FB FO	001E1 001E4	CALLS INSV	#1, CLISPRESENT R0, #2, #1, INSSGL_CTLMSK	0545
			50	AA	9F 001E9	PUSHAB	P.ABK	
69	01	68 03	01 50	FB FO	001EC 001EF	CALLS INSV	#1, CLISPRESENT R0, #3, #1, INSSGL_CTLMSK	0546
			64	AA	9F 001F4	PUSHAB	P.ABM	
02	A9	68 00	01 50	FB FO	001F7 001FA	CALLS INSV	#1, CLISPRESENT R0, #0, #1, INSSGL_CTLMSK+2	0547
			70	AA	9F 00200	PUSHAB	P.ABO	
01	A9	68 05	01 50	FB FO	00203 00206	CALLS INSV	#1, CLISPRESENT R0, #5, #1, INSSGL_CTLMSK+1	0548
			0088	CA	9F 0020C	PUSHAB	P.ABQ	
01	A9	68 06	01 50	FB FO	00210 00213	CALLS INSV	#1, CLISPRESENT R0, #6, #1, INSSGL_CTLMSK+1	0549
			88	AA	9F 00219	PUSHAB	PRIVILEGED DSC	
01	A9	68 07	01 50	FB FO	0021C 0021F	CALLS INSV	#1, CLISPRESENT R0, #7, #1, INSSGL_CTLMSK+1	0550
			0098	CA	9F 00225	PUSHAB	P.ABS	
02	A9	68 01	01 50	FB FO	00229 0022C	CALLS INSV	#1, CLISPRESENT R0, #1, #1, INSSGL_CTLMSK+2	0551
			00AC	CA	9F 00232	PUSHAB	P.ABU	
02	A9	68 02	01 50	FB FO	00236 00239	CALLS INSV	#1, CLISPRESENT R0, #2, #1, INSSGL_CTLMSK+2	0552
			00BC	CA	9F 0023F	PUSHAB	P.ABW	
02	A9	68 03	01 50	FB FO	00243 00246	CALLS INSV	#1, CLISPRESENT R0, #3, #1, INSSGL_CTLMSK+2	0553
			00D0	CA	9F 0024C	PUSHAB	P.ABY	
02	A9	68 05	01 50	FB FO	00250 00253	CALLS INSV	#1, CLISPRESENT R0, #5, #1, INSSGL_CTLMSK+2	0558
		0000V	CF	00	FB 00259	CALLS	#C, PROCESS_COMMAND	
			09	50	E8 0025E	BLBS	STATUS, 22\$	
		00000000G	00	50	DD 00261	PUSHL	STATUS	
			57	01	FB 00263	CALLS	#1, LIBSSIGNAL	
			03	08	AC DO 0026A	MOVL	INTERACTIVE, R7	0564
				57	E9 0026E	BLBC	R7, 23\$	
				FDB7	31 00271	BRW	1\$	
			50	01	DO 00274	MOVL	#1, R0	0565
					04 00277	RET		0566

INSPRSOLD  
V04-000

INSPARSE\_OLD\_CMD

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

; 370 0567 1

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)

IN  
VO

:

.....

.....

.....

.....

.....

.....

.....

.....

.....



```

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], ! arguement 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 .INSSGL_CTLMSK [INSSV_PRIV]
: 426 0622 2 THEN
: 427 0623 2 BEGIN
: 428 0624 2 LOCAL

```

```

process_command
: 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$FILL (0, 8, INSS$Q_KFEPRIVS); ! initialize privilege quadword to z
: 433      0629      3          WHILE CLISGET_VALUE (~PRIVILEGED_DSC, PRIV_DSC ) DO
: 434      0630      4              BEGIN
: 435      0631      4                  ALL PRIVS = FALSE; ! turn off default
: 436      0632      4                  SELECT PRV$SETPRIV (PRIV_DSC, INSS$Q_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$Q_KFEPRIVS); ! use default of all privs set
: 454      0650      3          END;
: 455      0651      2
: 456      0652      2      !+++
: 457      0653      2      :
: 458      0654      2      Act on the command entered
: 459      0655      2      :
: 460      0656      2      !---
: 461      0657      2
: 462      0658      3      IF (NOT .INSS$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

```



			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	
			6B		01	FB	0003C		MOVZWL	#10244, -(SP)	0617	
					55	11	0003F		CALLS	#1, LIBSSIGNAL		
				01	A7	95	00041	1\$:	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, (SPT), #0, #8, INSSGQ_KFEPRIVS	0628	
					5A	DD	0004F	2\$:	PUSHL	R10	0629	
				0000'	CF	9F	00051		PUSHAB	PRIVILEGED_DSC		
			6B		02	FB	00055		CALLS	#2, CLISGET_VALUE		
			3D		50	E9	00058		BLBC	R0, 6\$		
					56	D4	0005B		CLRL	ALL_PRIVS	0631	
					59	DD	0005D		PUSHL	R9	0632	
					5A	DD	0005F		PUSHL	R10		
		00000000G	00		02	FB	00061		CALLS	#2, PRV\$SETPRIV		
			52		50	D0	00068		MOVL	R0, R2		
		00000000G	8F		52	D1	0006B		CMPL	R2, #PRV\$_INVMAM	0635	
					0C	12	00072		BNEQ	3\$		
					5A	DD	00074		PUSHL	R10	0637	
					01	DD	00076		PUSHL	#1		
				00000000G	8F	DD	00078		PUSHL	#INSS_INVPRVAM		
					13	11	0007E		BRB	4\$		
		00000000G	8F		52	D1	00080	3\$:	CMPL	R2, #PRV\$_NOTUNG	0641	
					C6	12	00087		BNEQ	2\$		
					5A	DD	00089		PUSHL	R10	0643	
					01	DD	0008B		PUSHL	#1		
				00000000G	8F	DD	0008D		PUSHL	#INSS_PRVNOTUNG		
			6B		03	FB	00093	4\$:	CALLS	#3, LIBSSIGNAL		
					51	11	00096	5\$:	BRB	11\$	0644	
			07		56	E9	00098	6\$:	BLBC	ALL_PRIVS, 7\$	0648	
08	FF	8F	6E		00	2C	0009B		MOVCS	#0, -(SP), #-1, #8, INSSGQ_KFEPRIVS	0649	
					69		000A1					
			46		67	E8	000A2	7\$:	BLBS	INSSGL_CTLMSK, 12\$	0658	
				05	01	A7			BBS	#1, INSSGL_CTLMSK+1, 8\$	0668	
				17	01	A7			BBC	#2, INSSGL_CTLMSK+1, 9\$	0669	
					7E	D4	000AF	8\$:	CLRL	-(SP)	0672	
		00000000G	00		01	FB	000B1		CALLS	#1, INSSLIST		
			52		50	D0	000B8		MOVL	R0, STATUS		
			5F		52	E8	000BB		BLBS	STATUS, 15\$	0673	
					52	DD	000BE		PUSHL	STATUS	0676	
					CF	9F	000C0		PUSHAB	LIST_DSC	0675	
					18	11	000C4		BRB	10\$		
		52		01	A7	04	E1	000C6	9\$:	BBC	#4, INSSGL_CTLMSK+1, 15\$	0680
		00000000G	00		00	FB	000CB		CALLS	#0, INSSGLOBAL	0683	
			52		50	D0	000D2		MOVL	R0, STATUS		



dispatch

```
0725 1 %SBTTL 'dispatch';
0726 1 ROUTINE DISPATCH =
0727 1 |+++
0728 1 |
0729 1 |---
0730 1 |
0731 2 BEGIN
0732 2 LOCAL
0733 2 STATUS;
0734 2
0735 2 IF NOT .INSSG_KFEFAB[FAB$L_STS]
0736 2 AND NOT .INSSGL_CTLMSK[INSSV_DELETE]
0737 2 THEN RETURN TRUE;
0738 2 IF .INSSGL_CTLMSK [INSSV_DELETE]
0739 2 THEN
0740 2 |
0741 2 | if /delete requested then /replace/create are invalid
0742 2 | if /list requested then all the remaining known files will
0743 2 | be listed after the delete
0744 2 |
0745 2 BEGIN
0746 2 STATUS = INSSDELETE (); ! delete it
0747 2 IF NOT .STATUS
0748 2 THEN
0749 2 BEGIN
0750 2 SIGNAL (((INSS_FAIL AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
0751 2 2, _DELETE_DSC, INSSGQ_KFERNS, .STATUS);
0752 2
0753 2 IF .STATUS EQL INSS_INTRNLERR
0754 2 THEN
0755 2 SIGNAL (INSS_INTRNLERR, 1, .INSSL_INTRNLERR);
0756 2 END
0757 2 ELSE IF .STATUS EQL INSS_REMOVED
0758 2 THEN SIGNAL(.STATUS, T, INSSGQ_KFERNS);
0759 2 IF (.INSSGL_CTLMSK [INSSV_LIST] OR .INSSGL_CTLMSK [INSSV_FULL])
0760 2 THEN REPORT-(INSSLIST (0));
0761 2 END
0762 2
0763 2
0764 2 ELSE
0765 2 IF .INSSGL_CTLMSK [INSSV_REPLACE]
0766 2 THEN
0767 2 |
0768 2 | Replace has been requested. There can be no other qualifiers.
0769 2 | The entry must be deleted and recreated.
0770 2 | If /list requested with /replace then do the replace and
0771 2 | list the new entry.
0772 2 |
0773 2 BEGIN
0774 2 |
0775 2 | Delete and recreate with new attributes
0776 2 |
0777 2 | INSSGL_CTLMSK [INSSV_DELETE] = TRUE; ! set delete flag
0778 2 | INSSGL_REPLACE_MSK = 0;
0779 2 | STATUS = INSSDELETE (); ! recreate it
0780 2 | IF NOT .STATUS
0781 2 |
```

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
    INSSCLOSE_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 = INSS_EXISTS;

    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

```







		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		CPL	STATUS, R5		0812
				09	12	0C13A		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		CPL	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

C  
C  
C

INSPRSOLD  
V04-000

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)

\*\*F

0D	01	A3	02	E1	001CF	BBC	#2, INSSGL CTLMSK+1, 24\$	:	
		6A	67	DD	001D4	22\$:	PUSHL	INSSGL KFEADR	: 0869
		05	01	FB	001D6		CALLS	#1, INSSLIST	:
		64	50	E8	001D9		BLBS	STATUS, 24\$	:
		A3	50	DD	001DC	23\$:	PUSHL	STATUS	:
11	01		01	FB	001DE		CALLS	#1, LIBSSIGNAL	:
		00	04	E1	001E1	24\$:	BBC	#4, INSSGL CTLMSK+1, 26\$	: 0870
		05	67	DD	001E6		PUSHL	INSSGL KFEADR	: 0871
00000000G		05	01	FB	001E8		CALLS	#1, INSSGLOBAL	:
		64	50	E8	001EF	25\$:	BLBS	STATUS, 26\$	:
		50	50	DD	001F2		PUSHL	STATUS	:
			01	FB	001F4		CALLS	#1, LIBSSIGNAL	:
			01	DD	001F7	26\$:	MOVL	#1, R0	: 0873
			04	001FA			RET		: 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 \$SPLITS,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 : 0878
0E DD 00002 PUSHL #14 : 0899
00000000G 00 9F 00004 PUSHAB LIB$GET_INPUT : 0901
04 AE 9F 0000A PUSHAB HLP_FLAGS
0000' CF 9F 0000D PUSHAB P.ACA
04 AC DD 00011 PUSHL HLP_DSC
7E D4 00014 CLRL -(SP)
00000000G 00 9F 00016 PUSHAB LIB$PUT_OUTPUT
00000000G 00 06 FB 0001C CALLS #6, LBR$OUTPUT_HELP
09 50 E8 00023 BLBS STATUS, 1$
00000000G 00 50 DD 00026 PUSHL STATUS
01 FB 00028 CALLS #1, LIB$SIGNAL
09 01 D0 0002F 1$: MOVL #1, R0 : 0902
04 00032 RET : 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)

XF1  
V04

: 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)
\$SPLITS	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	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$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



The image displays a grid of 144 small terminal window screenshots, arranged in 12 rows and 12 columns. Each window shows a different command-line interface or data display from the VAX/VMS system. The windows are densely packed and contain various text-based outputs, including command prompts, error messages, and data listings. Some windows are more legible than others, showing clear text and structure. The overall appearance is that of a comprehensive manual or reference guide for the system's command-line utilities.

Visible window titles and content include:

- INSMAN LIS
- INDELETE LIS
- INSGLOBAL LIS
- INSKFSAN LIS
- INSLIST LIS
- INSOLDCMD LIS
- INSPRSOLD LIS
- INMSG LIS



