

CCCCCCCCCCCC	DDDDDDDDDDDD	UUU	UUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUU	UUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU


```
1 0001 0 MODULE listing (IDENT='V04-000',
2 0002 0 ADDRESSING_MODE(EXTERNAL=GENERAL))
3 0003 1 = BEGIN
4 0004 1
5 0005 1 !*****
6 0006 1 !*
7 0007 1 !* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
8 0008 1 !* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
9 0009 1 !* ALL RIGHTS RESERVED.
10 0010 1 !*
11 0011 1 !* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
12 0012 1 !* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
13 0013 1 !* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
14 0014 1 !* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
15 0015 1 !* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
16 0016 1 !* TRANSFERRED.
17 0017 1 !*
18 0018 1 !* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
19 0019 1 !* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
20 0020 1 !* CORPORATION.
21 0021 1 !*
22 0022 1 !* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
23 0023 1 !* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
24 0024 1 !*
25 0025 1 !*
26 0026 1 !*****
27 0027 1
28 0028 1 ++
29 0029 1 Facility: Command Definition Utility, Listing Generation
30 0030 1
31 0031 1 Abstract: This module is responsible for generating a listing of the
32 0032 1 CLD files we are compiling. All files are listed in one
33 0033 1 listing file, with headings, line numbers, and error messages.
34 0034 1
35 0035 1 Environment: Standard CDU environment.
36 0036 1
37 0037 1 Author: Paul C. Anagnostopoulos
38 0038 1 Creation: 1 December 1982
39 0039 1
40 0040 1 Modifications:
41 0041 1 --
42 0042 1
43 0043 1
44 0044 1 library 'sys$library:lib';
45 0045 1 require 'cdureq';
```

```

: 47      0459 1  !      T A B L E   O F   C O N T E N T S
: 48      0460 1  !      -----
: 49      0461 1  !
: 50      0462 1  forward routine
: 51      0463 1      cdu$prepare_listing_file: novalue,
: 52      0464 1      cdu$reject_listing_page: novalue,
: 53      0465 1      cdu$report_listing_line: novalue,
: 54      0466 1      cdu$report_listing_trailer: novalue;
: 55      0467 1
: 56      0468 1
: 57      0469 1  !      E X T E R N A L   R E F E R E N C E S
: 58      0470 1  !      -----
: 59      0471 1
: 60      0472 1  external routine
: 61      0473 1      cdu$report_listing_heading2,
: 62      0474 1      cdu$report_rms_error,
: 63      0475 1      cli$get_value,
: 64      0476 1      cli$present,
: 65      0477 1      lib$lp_lines,
: 66      0478 1      str$trim;
: 67      0479 1
: 68      0480 1  external
: 69      0481 1      cdu$facility_string: descriptor,
: 70      0482 1      cdu$gl_cld_errors: long;
: 71      0483 1
: 72      P 0484 1  $shr_msgdef(cdu,17,local,
: 73      P 0485 1      (openout,severe),
: 74      P 0486 1      (writeerr,error)
: 75      0487 1      );
```

```

77      0488 1  !      L I S T I N G   F I L E   C O N T R O L   B L O C K S
78      0489 1  !
79      0490 1  !
80      0491 1  ! The following items define the RMS control blocks needed to create and
81      0492 1  ! generate the listing file.
82      0493 1  !
83      0494 1  own
84      P 0495 1      listing_nam: $nam(
85      0496 1          ),
86      0497 1
87      0498 1      dbuffer(listing_spec,nam$c_maxrss),
88      P 0499 1      listing_fab: $fab(
89      P 0500 1          dnm='.LIS',
90      P 0501 1          fna=listing_spec+8,
91      P 0502 1          fns=%allocation(listing_spec)-8,
92      P 0503 1          fac=put,
93      P 0504 1          fop=<sqo,nam,ofp>,
94      P 0505 1          nam=listing_nam,
95      P 0506 1          org=seq,
96      P 0507 1          rat=cr,
97      P 0508 1          rfm=var
98      0509 1          ),
99      0510 1
100     P 0511 1      listing_rab: $rab(
101     P 0512 1          fab=listing_fab,
102     P 0513 1          rac=seq,
103     P 0514 1          rop=wbh
104     0515 1          );
105     0516 1
106     0517 1  !      L I S T I N G   C O N T R O L
107     0518 1  !
108     0519 1  !
109     0520 1  ! The following items keep track of the page number and the line count within
110     0521 1  ! a page.
111     0522 1  !
112     0523 1  own
113     0524 1      page_number: long,
114     0525 1      line_count: long;

```

```

116 0526 1 !++
117 0527 1 ! Description: This routine is called to prepare the listing file for the
118 0528 1 ! next CLD file listing.
119 0529 1
120 0530 1 ! Parameters: cld_fab By reference, the FAB for the new CLD file.
121 0531 1
122 0532 1 ! Returns: Nothing.
123 0533 1
124 0534 1 ! Notes: The listing file is created on the first call.
125 0535 1 !--
126 0536 1
127 0537 1 GLOBAL ROUTINE cdu$prepare_listing_file(cld_fab: pointer) : novalue
128 0538 2 = BEGIN
129 0539 2
130 0540 2 local
131 0541 2 status: long;
132 0542 2
133 0543 2
134 0544 2 ! Determine if we are to generate a listing file.
135 0545 2
136 0546 3 if cli$present(dtext('LISTING')) then (
137 0547 3
138 0548 3 ! If we have not already created the listing file, do so.
139 0549 3
140 0550 4 if .listing_fab[fab$w_ifi] eqlu 0 then (
141 0551 4
142 0552 4 ! Get any value specified on the listing qualifier to use as
143 0553 4 ! the spec for the listing file.
144 0554 4
145 0555 4 cli$get_value(dtext('LISTING'),listing_spec);
146 0556 4
147 0557 4 ! Use the CLD file NAM block as the related NAM block for our
148 0558 4 ! listing file.
149 0559 4
150 0560 4 listing_nam[nam$l_rlf] = .cld_fab[fab$l_nam];
151 0561 4
152 0562 4 ! Create and connect to the listing file.
153 0563 4
154 0564 4 status = $create(fab=listing_fab);
155 0565 5 if .status then (
156 0566 5 status = $connect(rab=listing_rab);
157 0567 5 if not .status then
158 0568 5 cdu$report_rms_error(msg(cdu$_openout),listing_rab);
159 0569 4 ) else
160 0570 4 cdu$report_rms_error(msg(cdu$_openout),listing_fab);
161 0571 3 );
162 0572 3
163 0573 3 ! Prepare for a new listing by resetting the page number and
164 0574 3 ! ejecting the page.
165 0575 3
166 0576 3 page_number = 0;
167 0577 3 cdu$eject_listing_page();
168 0578 2 );
169 0579 2
170 0580 2 return;
171 0581 2
172 0582 1 END;

```

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

.PSECT $SPLITS,NOWRT,NOEXE,2

00 47 4E 49 53 49 4C 2E 00000 P.AAA: .ASCII \.LIS\
00 47 4E 49 54 53 49 4C 00004 P.AAC: .ASCII \LISTING\<0>
010E0007 0000C P.AAB: .LONG 17694727
00000000' 00010 .ADDRESS P.AAC
00 47 4E 49 54 53 49 4C 00014 P.AAE: .ASCII \LISTING\<0>
010E0007 0001C P.AAD: .LONG 17694727
00000000' 00020 .ADDRESS P.AAE

.PSECT $OWNS,NOEXE,2

02 00000 LISTING_NAM:
00000000 .BYTE 2
60 00001 .BYTE 96
00 00002 .BYTE 0
00 00003 .BYTE 0
00000000 00004 .LONG 0
00 00008 .BYTE 0
00 00009 .BYTE 0
00 0000A .BYTE 0
00 0000B .BYTE 0
00000000 0000C .LONG 0
00000000 00010 .LONG 0
0000# 00014 .WORD 0[8]
0000# 00024 .WORD 0[3]
0000# 0002A .WORD 0[3]
00000000 00030 .LONG 0
00000000 00034 .LONG 0
00 00038 .BYTE 0
00 00039 .BYTE 0
00 0003A .BYTE 0
00 0003B .BYTE 0
00 0003C .BYTE 0
00 0003D .BYTE 0
00# 0003E .BYTE 0[2]
00000000 00040 .LONG 0
00000000 00044 .LONG 0
00000000 00048 .LONG 0
00000000 0004C .LONG 0
00000000 00050 .LONG 0
00000000 00054 .LONG 0
00000000# 00058 .LONG 0[2]
00FF 00060 LISTING_SPEC:
00 00 00062 .WORD 255
00000000' 00064 .BYTE 0, 0
00068 .ADDRESS LISTING_SPEC+8
00167 .BLKB 255
03 00168 LISTING_FAB:
00169 .BLKB 1
00168 .BYTE 3
50 00169 .BYTE 80
```

0000	0016A	.WORD	0
21000040	0016C	.LONG	553648192
00000000	00170	.LONG	0
00000000	00174	.LONG	0
00000000	00178	.LONG	0
0000	0017C	.WORD	0
01	0017E	.BYTE	1
00	0017F	.BYTE	0
00000000	00180	.LONG	0
00	00184	.BYTE	0
00	00185	.BYTE	0
02	00186	.BYTE	2
02	00187	.BYTE	2
00000000	00188	.LONG	0
00000000	0018C	.LONG	0
00000000	00190	.ADDRESS	LISTING_NAM
00000000	00194	.ADDRESS	LISTING_SPEC+8
00000000	00198	.ADDRESS	P.AAA
FF	0019C	.BYTE	-1
04	0019D	.BYTE	4
0000	0019E	.WORD	0
00000000	001A0	.LONG	0
0000	001A4	.WORD	0
00	001A6	.BYTE	0
00	001A7	.BYTE	0
00000000	001A8	.LONG	0
00000000	001AC	.LONG	0
0000	001B0	.WORD	0
00	001B2	.BYTE	0
00	001B3	.BYTE	0
00000000	001B4	.LONG	0
01	001B8	LISTING_RAB:	
		.BYTE	1
44	001B9	.BYTE	68
0000	001BA	.WORD	0
00000400	001BC	.LONG	1024
00000000	001C0	.LONG	0
00000000	001C4	.LONG	0
0000	001C8	.WORD	0[3]
0000	001CE	.WORD	0
00000000	001D0	.LONG	0
0000	001D4	.WORD	0
00	001D6	.BYTE	0
00	001D7	.BYTE	0
0000	001D8	.WORD	0
0000	001DA	.WORD	0
00000000	001DC	.LONG	0
00000000	001E0	.LONG	0
00000000	001E4	.LONG	0
00000000	001E8	.LONG	0
00	001EC	.BYTE	0
00	001ED	.BYTE	0
00	001EE	.BYTE	0
00	001EF	.BYTE	0
00000000	001F0	.LONG	0
00000000	001F4	.ADDRESS	LISTING_FAB
00000000	001F8	.LONG	0

001FC PAGE_NUMBER:
.BLKB 4
00200 LINE_COUNT:
.BLKB 4

.EXTRN CDUSREPORT_LISTING_HEADING2
.EXTRN CDUSREPORT_RMS_ERROR
.EXTRN CLISGET_VALUE, CLISPRESENT
.EXTRN LIB\$LP_LINES, STR\$TRIM
.EXTRN CDUSFACILITY_STRING
.EXTRN CDUSGL_CLD_ERRORS
.EXTRN SYSS\$CREATE, SYSS\$CONNECT

.PSECT \$CODE\$,NOWRT,2

			0004	00000	.ENTRY	CDUSPREPARE_LISTING_FILE, Save R2	:	0537
	52	0000'	CF	9E 00002	MOVAB	LISTING_FAB, R2	:	
		0000'	CF	9F 00007	PUSHAB	P.AAB	:	0546
00000000G	00		01	FB 0000B	CALLS	#1, CLISPRESENT	:	
	54		50	E9 00012	BLBC	R0, 4\$:	
			02	A2 B5 00015	TSTW	LISTING_FAB+2	:	0550
			46	12 00018	BNEQ	3\$:	
		FEF8	C2	9F 0001A	PUSHAB	LISTING_SPEC	:	0555
		0000'	CF	9F 0001E	PUSHAB	P.AAD	:	
00000000G	00		02	FB 00022	CALLS	#2, CLISGET_VALUE	:	
	50		04	AC D0 00029	MOVL	CLD FAB, R0	:	0560
FEAB	C2		28	A0 D0 0002D	MOVL	40(R0), LISTING_NAM+16	:	
			52	DD 00033	PUSHL	R2	:	0564
00000000G	00		01	FB 00035	CALLS	#1, SYSS\$CREATE	:	
	12		50	E9 0003C	BLBC	STATUS, 1\$:	0565
			50	A2 9F 0003F	PUSHAB	LISTING_RAB	:	0566
00000000G	00		01	FB 00042	CALLS	#1, SYSS\$CONNECT	:	
	14		50	E8 00049	BLBS	STATUS, 3\$:	0567
			50	A2 9F 0004C	PUSHAB	LISTING_RAB	:	0568
			02	11 0004F	BRB	2\$:	
			52	DD 00051 1\$:	PUSHL	R2	:	0570
		001110A4	8F	DD 00053 2\$:	PUSHL	#1118372	:	
00000000G	00		02	FB 00059	CALLS	#2, CDUSREPORT_RMS_ERROR	:	
		0094	C2	D4 00060 3\$:	CLRL	PAGE_NUMBER	:	0576
0000V	CF		00	FB 00064	CALLS	#0, CDUSEJECT_LISTING_PAGE	:	0577
			04	00069 4\$:	RET		:	0582

; Routine Size: 106 bytes, Routine Base: \$CODE\$ + 0000

			000C 00000	.PSECT	\$CODE\$,NOWRT,2	
	53	0000V	CF 9E 00002	.ENTRY	CDUSEJECT LISTING_PAGE, Save R2,R3	: 0594
	52	00000000G	8F D0 00007	MOVAB	CDUSREPORT_LISTING_LINE, R3	
		0000'	CF D6 0000E	MOVL	#CDUS ANYTHING, R2	
00000000G	00		00 FB 00012	INCL	PAGE_NUMBER	: 0600
0000'	CF	FB	A0 9E 00019	CALLS	#0, [IB\$LP LINES	: 0601
		0000'	CF 9F 0001F	MOVAB	-5(R0), LINE_COUNT	
		00010001	8F DD 00023	PUSHAB	P.AAF	: 0605
			52 DD 00029	PUSHL	#65537	
	63		03 FB 0002B	PUSHL	R2	
		0000'	CF DD 0002E	CALLS	#3, CDUSREPORT_LISTING_LINE	
		00000000G	00 9F 00032	PUSHL	PAGE_NUMBER	: 0609
			7E D4 00038	PUSHAB	CDUSFACILITY_STRING	
		00010003	8F DD 0003A	CLRL	-(SP)	
		00000000G	8F DD 00040	PUSHL	#65539	
	63		05 FB 00046	PUSHL	#CDUS HEADING1	
00000000G	00		00 FB 00049	CALLS	#5, CDUSREPORT_LISTING_LINE	
		0000'	CF 9F 00050	CALLS	#0, CDUSREPORT_LISTING_HEADING2	: 0614
		00010001	8F DD 00054	PUSHAB	P.AAH	: 0618
			52 DD 0005A	PUSHL	#65537	
	63		03 FB 0005C	PUSHL	R2	
		0000'	CF 9F 0005F	CALLS	#3, CDUSREPORT_LISTING_LINE	
		00010001	8F DD 00063	PUSHAB	P.AAJ	: 0619
			52 DD 00069	PUSHL	#65537	
	63		03 FB 0006B	PUSHL	R2	
			04 0006E	CALLS	#3, CDUSREPORT_LISTING_LINE	: 0623
				RET		

: Routine Size: 111 bytes, Routine Base: \$CODE\$ + 006A

```

: 216 0624 1 |++
: 217 0625 1 | Description: This routine is called to generate a line in the listing file.
: 218 0626 1 | The line is created using $FAO and written into the listing
: 219 0627 1 | file without any other manipulation.
: 220 0628 1 |
: 221 0629 1 | Parameters: A standard $PUTMSG argument list.
: 222 0630 1 |
: 223 0631 1 | Returns: Nothing.
: 224 0632 1 |
: 225 0633 1 | Notes:
: 226 0634 1 | --
: 227 0635 1 |
: 228 0636 1 | GLOBAL ROUTINE cdu$report_listing_line(message: long,
: 229 0637 1 | fao_count: word,
: 230 0638 1 | faoT: long) : novalue
: 231 0639 2 | = BEGIN
: 232 0640 2 |
: 233 0641 2 | builtin
: 234 0642 2 | argptr;
: 235 0643 2 |
: 236 0644 2 | ! The following little routine is the action routine for $PUTMSG. It
: 237 0645 2 | ! receives a formatted line, which it writes into the listing file.
: 238 0646 2 |
: 239 0647 2 | ROUTINE write_line(line: ref descriptor)
: 240 0648 2 | = BEGIN
: 241 0649 2 |
: 242 0650 2 | local
: 243 0651 2 | status: long;
: 244 0652 2 |
: 245 0653 2 | listing_rab[rab$w_rsz] = .line[len];
: 246 0654 2 | listing_rab[rab$l_rbf] = .line[ptr];
: 247 0655 2 | status = $put(rab=listing_rab);
: 248 0656 2 | if not .status then
: 249 0657 2 | cdu$report_rms_error(msg(cdu$_writeerr),listing_rab);
: 250 0658 2 |
: 251 0659 2 | return false;
: 252 0660 2 |
: 253 0661 2 | END;

```

.EXTRN SYSSPUT

```

0004 0000 WRITE_LINE:
      .WORD Save R2 : 0647
      MOVAB LISTING_RAB, R2 : 0653
      MOVL LINE, R0
      MOVW (R0), LISTING_RAB+34 : 0654
      MOVL 4(R0), LISTING_RAB+40 : 0655
      PUSHL R2
      CALLS #1, SYSSPUT
      BLBS STATUS, 1$ : 0656
      PUSHL R2 : 0657
      PUSHL #1118418
      CALLS #2, CDU$REPORT_RMS_ERROR
      CLRL R0 : 0659
      RET : 0661

```

; Routine Size: 50 bytes, Routine Base: \$CODE\$ + 00D9

```

: 254      0662 2
: 255      0663 2
: 256      0664 2 ! If a listing file is not being generated, just return.
: 257      0665 2
: 258      0666 2 if .listing_fab[fab$w_ifi] eqlu 0 then
: 259      0667 2     return;
: 260      0668 2
: 261      0669 2 ! If there are no lines left on the current page, then eject it.
: 262      0670 2
: 263      0671 2 if .line_count eqlu 0 then
: 264      0672 2     cdu$eject_listing_page();
: 265      0673 2
: 266      0674 2 ! Use $PUTMSG to format the message lines. It will call our action routine
: 267      0675 2 ! to write them into the listing file.
: 268      0676 2
: 269      P 0677 2 $putmsg(msgvec=argptr(),
: 270      0678 2     actrtn=write_line);
: 271      0679 2
: 272      0680 2 ! Account for the current line.
: 273      0681 2
: 274      0682 2 decrement(line_count);
: 275      0683 2 return;
: 276      0684 2
: 277      0685 1 END;

```

```

                                .EXTRN SYSS$PUTMSG
                                .ENTRY  CDU$REPORT LISTING_LINE, Save nothing      : 0636
                                TSTW    LISTING_FAB+2                             : 0666
                                BEQL    2$
                                TSTL    LINE_COUNT                               : 0671
                                BNEQ    1$
                                CALLS   #0, CDUSEJECT_LISTING_PAGE              : 0672
                                CLRQ    -(SP)                                     : 0678
                                PUSHAB  WRITE_LINE
                                PUSHL   AP
                                CALLS   #4, SYSS$PUTMSG
                                DECL    LINE_COUNT                               : 0682
                                RET

```

; Routine Size: 38 bytes, Routine Base: \$CODE\$ + 010B

```

: 279      0686 1 |++
: 280      0687 1 | Description: This routine is called after a CLD file has been completely
: 281      0688 1 | listed. It generates a nice little trailer in the listing
: 282      0689 1 | file, which includes the error count and the command line
: 283      0690 1 | that invoked the CDU.
: 284      0691 1 |
: 285      0692 1 | Parameters: None.
: 286      0693 1 |
: 287      0694 1 | Returns: Nothing.
: 288      0695 1 |
: 289      0696 1 | Notes:
: 290      0697 1 | --
: 291      0698 1 |
: 292      0699 1 GLOBAL ROUTINE cdu$report_listing_trailer          : novalue
: 293      0700 2 = BEGIN
: 294      0701 2
: 295      0702 2
: 296      0703 2 ! Skip down a couple of lines and report the error count, unless it's zero.
: 297      0704 2
: 298      0705 2 cdu$report_listing_line(msg(cdu$_anything),nobabble+1,dtext(''));
: 299      0706 2 if .cdu$gl_cld_errors nequ 0 then (
: 300      0707 2     cdu$report_listing_line(msg(cdu$_anything),nobabble+1,dtext(''));
: 301      0708 2     cdu$report_listing_line(msg(cdu$_errorcount),nobabble+1,.cdu$gl_cld_errors);
: 302      0709 2 );
: 303      0710 2
: 304      0711 2 ! Skip down another line and report the command line.
: 305      0712 2
: 306      0713 2 cdu$report_listing_line(msg(cdu$_anything),nobabble+1,dtext(''));
: 307      0714 2 with_dbuffer(command_line,1024,
P 0715 2     cli$get_value(dtext('$LINE'),command_line);
P 0716 2     str$trim(command_line,command_line,command_line);
P 0717 2     cdu$report_listing_line(msg(cdu$_commandline),nobabble+1,command_line);
: 311      0718 2 );
: 312      0719 2
: 313      0720 2 return;
: 314      0721 2
: 315      0722 1 END;

```

.PSECT \$SPLITS,NOWRT,NOEXE,2

	010E0000	00040	P.AAM:	.BLKB	0
	00000000	00040	P.AAL:	.LONG	17694720
		00044		.ADDRESS	P.AAM
	010E0000	00048	P.AAO:	.BLKB	0
	00000000	00048	P.AAN:	.LONG	17694720
		0004C		.ADDRESS	P.AAO
	010E0000	00050	P.AAQ:	.BLKB	0
	00000000	00050	P.AAP:	.LONG	17694720
		00054		.ADDRESS	P.AAQ
00 00 00 45 4E 49 4C 24	010E0005	00058	P.AAS:	.ASCII	\\$LINE\<0><0><0>
	00000000	00060	P.AAR:	.LONG	17694725
		00064		.ADDRESS	P.AAS

.EXTRN CDU\$_ERRORCOUNT
.EXTRN CDU\$_COMMANDLINE

```

.PSECT $CODE$,NOWRT,2
.ENTRY CDUSREPORT_LISTING_TRAILER, Save R2,R3,R4 : 0699
MOVAB CDUSGL_CLD_ERRORS, R4
MOVAB CDUSREPORT_LISTING_LINE, R3
MOVL #CDUS_ANYTHING, R2
MOVAB -1032(TSP), SP
PUSHAB P.AAL : 0705
PUSHL #65537
PUSHL R2
CALLS #3, CDUSREPORT_LISTING_LINE
TSTL CDUSGL_CLD_ERRORS : 0706
BEQL 1$
PUSHAB P.AAN : 0707
PUSHL #65537
PUSHL R2
CALLS #3, CDUSREPORT_LISTING_LINE
PUSHL CDUSGL_CLD_ERRORS : 0708
PUSHL #65537
PUSHL #CDUS_ERRORCOUNT
CALLS #3, CDUSREPORT_LISTING_LINE
PUSHAB P.AAP : 0713
PUSHL #65537
PUSHL R2
CALLS #3, CDUSREPORT_LISTING_LINE
MOVZWL #1024, COMMAND_LINE : 0718
MOVAB COMMAND_LINE+8, COMMAND_LINE+4
PUSHL SP
PUSHAB P.AAR
CALLS #2, CLISGET_VALUE
PUSHL SP
PUSHAB COMMAND_LINE
PUSHAB COMMAND_LINE
CALLS #3, STRSTRIM
PUSHL SP
PUSHL #65537
PUSHL #CDUS_COMMANDLINE
CALLS #3, CDUSREPORT_LISTING_LINE : 0722
RET

```

; Routine Size: 147 bytes, Routine Base: \$CODE\$ + 0131

```

: 316      0723 1
: 317      0724 1 END
: 318      0725 0 ELUDOM

```

PSECT SUMMARY

```

:
: Name          Bytes          Attributes
:

```

LISTING
V04-000

1 2
15-Sep-1984 23:42:47
14-Sep-1984 11:58:24

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[CDU.SRC]LISTING.B32;1 Page 14
(7)

```
: SOWNS      516 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
: SPLITS    104 NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
: $CODES    452 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	54	0	1000	00:01.8

COMMAND QUALIFIERS

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

```
: Size:      452 code + 620 data bytes
: Run Time:   00:14.4
: Elapsed Time: 00:45.3
: Lines/CPU Min: 3031
: Lexemes/CPU-Min: 27700
: Memory Used: 129 pages
: Compilation Complete
```

SYMBOLS LIS
NODES LIS
OBJECT LIS
PARSE1 LIS
PARSE3 LIS
ROUTINES LIS
LISTING LIS
MAIN LIS
PARSE2 LIS
TABLE LIS