SYSLOA

CSP
V04

```
  CCCCCCCC    SSSSSSSS  PPPPPPPP    CCCCCCCC LL         IIIIII  EEEEEEEEEE  NN      NN  TTTTTTTTTT
  CCCCCCCC    SSSSSSSS  PPPPPPPP    CCCCCCCC LL         IIIIII  EEEEEEEEEE  NN      NN  TTTTTTTTTT
CC            SS        PP      PP CC        LL           II    EE          NN      NN      TT
CC            SS        PP      PP CC        LL           II    EE          NNNN    NN      TT
CC            SS        PP      PP CC        LL           II    EE          NNNN    NN      TT
CC            SS        PP      PP CC        LL           II    EE          NNNN    NN      TT
CC              SSSSSS  PPPPPPPP  CC         LL           II    EEEEEEEE     NN  NN  NN      TT
CC              SSSSSS  PPPPPPP   CC         LL           II    EEEEEEEE     NN  NN  NN      TT
CC                  SS  PP        CC         LL           II    EE          NN      NN      TT
CC                  SS  PP        CC         LL           II    EE          NN    NNNN      TT
CC                  SS  PP        CC         LL           II    EE          NN      NN      TT
CC                  SS  PP        CC         LL           II    EE          NN      NN      TT
  CCCCCCCC  SSSSSSSS  PP          CCCCCCCC LLLLLLLLLL  IIIIII  EEEEEEEEEE  NN      NN      TT
  CCCCCCCC  SSSSSSSS  PP          CCCCCCCC LLLLLLLLLL  IIIIII  EEEEEEEEEE  NN      NN      TT
```

```
LL            IIIIII      SSSSSSSS
LL            III  I      SSSSSSSS
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
LLLLLLLLLL   IIIIII    SSSSSSSS
LLLLLLLLLL   IIIIII    SSSSSSSS
```
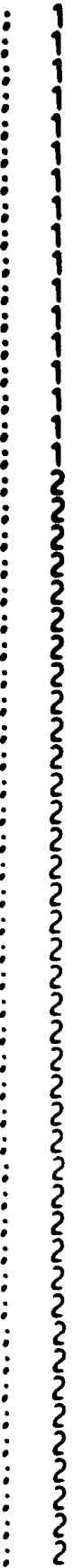
....
....
....
....

: R

:

```
0000     1              .TITLE   CSPCLIENT        - Client jacket routine calls to CSP
0000     2              .IDENT   'V04-000'
0000     3
0000     4      ;*******************************************************************************
0000     5      ;*                                                                             *
0000     6      ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                    *
0000     7      ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
0000     8      ;*  ALL RIGHTS RESERVED.                                                       *
0000     9      ;*                                                                             *
0000    10      ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED      *
0000    11      ;*  ONLY IN ACCORDANCE WITH THE  TERMS  OF  SUCH  LICENSE  AND WITH THE        *
0000    12      ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER      *
0000    13      ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY      *
0000    14      ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY      *
0000    15      ;*  TRANSFERRED.                                                               *
0000    16      ;*                                                                             *
0000    17      ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE      *
0000    18      ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT      *
0000    19      ;*  CORPORATION.                                                               *
0000    20      ;*                                                                             *
0000    21      ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS      *
0000    22      ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                    *
0000    23      ;*                                                                             *
0000    24      ;*                                                                             *
0000    25      ;*******************************************************************************
0000    26
0000    27      ;++
0000    28      ;
0000    29      ; FACILITY:     VMS
0000    30      ;
0000    31      ; ABSTRACT:     Various "clients" are placed here for loading with CLUSTERLOA.
0000    32      ;
0000    33      ; AUTHOR:       Jake VanNoy
0000    34      ;
0000    35      ; DATE:         27-May-1983
0000    36      ;
0000    37      ; REVISION HISTORY:
0000    38      ;
0000    39      ;       V03-003 ADE0001         Alan D. Eldridge        4-Feb-1984
0000    40      ;               Convert to use EXE$CSP_BRDCST.
0000    41      ;
0000    42      ;       V03-002 JLV0306         Jake VanNoy             29-AUG-1983
0000    43      ;               Check error status and deallocate CSD if failure.
0000    44      ;
0000    45      ;       V03-001 JLV0284         Jake VanNoy             28-JUL-1983
0000    46      ;               Minor changes to allow cluster writes.
0000    47      ;
0000    48      ;--
0000    49
0000    50
0000    51      ; Symbols
0000    52      ;
0000    53
0000    54              $BRKTDEF
0000    55              $CSDDEF
0000    56              $DYNDEF
0000    57
```

```
                     0000     58 ;
                     0000     59 ; Local data
                     0000     60 ;
                     0000     61
                 00000000     62          .PSECT  $$$190              ; Read Only Data
                     0000     63
                     0000     64
                 00000000     65          .PSECT  $$$200              ; Code
```

CSPCLIENT
V04-000
E 4
- Client jacket routine calls to CSP   16-SEP-1984 00:31:49 VAX/VMS Macro V04-00   Page 3
'EXE$CSP_BRKTHRU - Send Break-thru messa 5-SEP-1984 04:08:43 [SYSLOA.SRC]CSPCLIENT.MAR;1   (2)
CSPM
V04-

```
                              0000      67 .SBTTL   'EXE$CSP_BRKTHRU - Send Break-thru message throughout cluster'
                              0000      68 ;++
                              0000      69 ;
                              0000      70 ;
                              0000      71 ;  CALLING SEQUENCE:     JSB      EXE$CSP_BRKTHRU at IPL 0
                              0000      72 ;
                              0000      73 ;  INPUT PARAMETERS:     R6       Address of BRK structure
                              0000      74 ;
                              0000      75 ;  OUTPUT PARAMETERS:    R0,R1    Garbage
                              0000      76 ;
                              0000      77 ;                        All registers are preserved except R0,R1.
                              0000      78 ;
                              0000      79 ;  COMPLETION CODES:     R0       completion status:
                              0000      80 ;                                 SS$_NORMAL = normal success
                              0000      81 ;                                 SS$_...    = various error codes
                              0000      82 ;--
                              0000      83 EXE$CSP_BRKTHRU::
                              0000      84         ;
                              0000      85         ;
                              0000      86         ;     Get a CSD.
                              0000      87         ;
                              0000      88         ;
            57    008C C6 3C  0000      89         MOVZWL   BRK$W_MSGLEN(R6),R7          ; Message length
   51    57  00000072 8F C1  0005      90         ADDL3    #CSD$AB_DATA -               ; Add to CSD header
                              000D      91                  +CSD$T_BRK_MSGBUF,R7,R1
            00000000'GF 16    000D      92         JSB      G^EXE$ALLOC_CSD             ; Allocate and init a CSD
                   5E 50 E9   0013      93         BLBC     R0,200$                     ; If LBC, return error
                              0016      94         ;
                              0016      95         ;
                              0016      96         ;     Init private portion of CSD
                              0016      97         ;
                              0016      98         ;
                              0016      99         ASSUME   CSD$L_RECVOFF   EQ   4+CSD$L_RECVLEN
                              0016     100
               1A A2 7C       0016     101         CLRQ     CSD$L_RECVLEN(R2)           ; No return data this trip
               22 A2 D4       0019     102         CLRL     CSD$A_ASTADR(R2)           ; No AST needed -- return will
                              001C     103                                             ; therefore be synchronous
            33 A2 00' 90      001C     104         MOVB     S^#EXE$C_SYSEFN,CSD$B_EFN(R2) ; Legal event flag
            0E A2 01 CE       0020     105         MNEGL    #1,CSD$L_CSID(R2)           ; Use ALL BUT "my" node
            58 52 A2 9E       0024     106         MOVAB    CSD$AB_DATA(R2),R8          ; Get ptr to private data area
         16 A2 58 52 C3       0028     107         SUBL3    R2,R8,CSD$L_SENDOFF(R2)     ; Fill in the xmt data offset
         12 A2 57 20 C1       002D     108         ADDL3    #CSD$T_BRK_MSGBUF,R7,-      ; Fill in the xmt data length
                              0032     109                  CSD$L_SENDLEN(R2)
            0C A2 02 B0       0032     110         MOVW     #CSD$K_BRKTHRU,CSD$W_CODE(R2) ; Client code
                              0036     111         ;
                              0036     112         ;
                              0036     113         ;     Copy in BRKTHRU specific protocol
                              0036     114         ;
                              0036     115
                              0036     116         ASSUME   CSD$L_BRK_FLAGS    EQ   4+CSD$L_BRK_CARCON
                              0036     117         ASSUME   CSD$T_BRK_SENDTO   EQ   4+CSD$L_BRK_FLAGS
                              0036     118         ASSUME   CSD$W_BRK_SNDTYP   EQ   CSD$T_BRK_SENDTO+BRK$S_SENDNAME
                              0036     119         ASSUME   CSD$W_BRK_REQID    EQ   2+CSD$W_BRK_SNDTYP
                              0036     120         ASSUME   CSD$W_BRK_TIMOUT   EQ   2+CSD$W_BRK_REQID
                              0036     121         ASSUME   CSD$W_BRK_MSGLEN   EQ   2+CSD$W_BRK_TIMOUT
                              0036     122         ASSUME   CSD$T_BRK_MSGBUF   EQ   2+CSD$W_BRK_MSGLEN
                              0036     123
```

CSPCLIENT
V04-000
F 4
- Client jacket routine calls to CSP   16-SEP-1984 00:31:49 VAX/VMS Macro V04-00   Page   4
'EXE$CSP_BRKTHRU - Send Break-thru messa 5-SEP-1984 04:08:43 [SYSLOA.SRC]CSPCLIENT.MAR;1   (2)

```
                52   DD   0036  124          PUSHL   R2                              ; Save address of CSD
                          0038  125
          53    58   DO   0038  126          MOVL    R8,R3                           ; First address
       83   34  A6   DO   003B  127          MOVL    BRK$L_CARCON(R6),(R3)+          ; carriage control
       83   38  A6   DO   003F  128          MOVL    BRK$L_FLAGS(R6),(R3)+           ; Flags
                10   28   0043  129          MOVC3   #CSD$S_BRK_SENDTO,-
       63   3C  A6        0045  130                  BRK$T_SENDNAME(R6),(R3)         ; Copy in send to string
       83   4C  A6   BO   0048  131          MOVW    BRK$W_SENDTYPE(R6),(R3)+        ; Send type
       83   50  A6   BO   004C  132          MOVW    BRK$L_REQID(R6),(R3)+           ; Requestor ID (low word)
       83   4E  A6   BO   0050  133          MOVW    BRK$W_SECONDS(R6),(R3)+         ; Time out in seconds
          83  57   BO   0054  134            MOVW    R7,(R3)+                        ; Copy length
    63  008E C6  57   28   0057  135          MOVC3   R7,BRK$T_MSGBUF(R6),(R3)        ; and message data
                          005D  136
                52 8EDO   005D  137          POPL    R2                              ; Get address of CSD
                          0060  138          ;
                          0060  139          ;
                          0060  140          ;       Send the buffer to the target node.
                          0060  141          ;
                          0060  142          ;       Since we specified an AST address of 0, the call does not return
                          0060  143          ;       until the dialogue with the last node has completed. Furthermore,
                          0060  144          ;       this means that we should deallocate the CSD whether or not there's
                          0060  145          ;       an error.
                          0060  146          ;
                          0060  147          ;       See EXE$CSP_BRDCST preamble if an AST routine is to be added.
                          0060  148          ;       Also, if an AST routine is to be added, make sure that our caller
                          0060  149          ;       does not have AST's enabled.
                          0060  150          ;
                          0060  151          ;
    00000000'GF   16   0060  152            JSB     G^EXE$CSP_BRDCST                ; Send message out to world
                50   DD   0066  153          PUSHL   R0                              ; Save status
                          0068  154
          50   52   DO   0068  155          MOVL    R2,R0                           ; Input to deallocate routine
    00000000'GF   16   006B  156            JSB     G^EXE$DEALLOC_CSD               ; Done with it.
                          0071  157
                50 8EDO   0071  158          POPL    R0                              ; Recover status
                     05   0074  159  200$:   RSB                                     ; Done
                          0075  160
                          0075  161
                          0075  162  .END
```

```
BRK$L_CARCON       = 00000034
BRK$L_FLAGS        = 00000038
BRK$L_REQID        = 00000050
BRK$S_SENDNAME     = 00000010
BRK$T_MSGBUF       = 0000008E
BRK$T_SENDNAME     = 0000003C
BRK$W_MSGLEN       = 0000008C
BRK$W_SECONDS      = 0000004E
BRK$W_SENDTYPE     = 0000004C
CSD$AB_DATA        = 00000052
CSD$A_ASTADR       = 00000022
CSD$B_EFN          = 00000033
CSD$K_BRKTHRU      = 00000002
CSD$L_BRK_CARCON   = 00000000
CSD$L_BRK_FLAGS    = 00000004
CSD$L_CSID         = 0000000E
CSD$L_RECVLEN      = 0000001A
CSD$L_RECVOFF      = 0000001E
CSD$L_SENDLEN      = 00000012
CSD$L_SENDOFF      = 00000016
CSD$S_BRK_SENDTO   = 00000010
CSD$T_BRK_MSGBUF   = 00000020
CSD$T_BRK_SENDTO   = 00000008
CSD$W_BRK_MSGLEN   = 0000001E
CSD$W_BRK_REQID    = 0000001A
CSD$W_BRK_SNDTYP   = 00000018
CSD$W_BRK_TIMOUT   = 0000001C
CSD$W_CODE         = 0000000C
EXE$ALLOC_CSD        ********   X   03
EXE$CSP_BRDCST       ********   X   03
EXE$CSP_BRKTHRU      00000000  RG      03
EXE$C_SYSEFN         ********   X   03
EXE$DEALLOC_CSD      ********   X   03
```

+------------------+
! Psect synopsis !
+------------------+

PSECT name	Allocation		PSECT No.		Attributes											
. ABS .	00000000	(0.)	00	(0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS	00000000	(0.)	01	(1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
$$$190	00000000	(0.)	02	(2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
$$$200	00000075	(117.)	03	(3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

+--------------------------+
! Performance indicators !
+--------------------------+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:01.22
Command processing	112	00:00:00.40	00:00:01.85
Pass 1	187	00:00:02.15	00:00:08.67
Symbol table sort	0	00:00:00.25	00:00:01.62
Pass 2	47	00:00:00.46	00:00:02.74
Symbol table output	5	00:00:00.03	00:00:00.03

```
Psect synopsis output                2      00:00:00.02      00:00:00.02
Cross-reference output               0      00:00:00.00      00:00:00.00
Assembler run totals               384      00:00:03.37      00:00:16.15
```

The working set limit was 1050 pages.
16376 bytes (32 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 295 non-local and 1 local symbols.
162 source lines were read in Pass 1, producing 15 object records in Pass 2.
11 pages of virtual memory were used to define 10 macros.

```
                              +-----------------------------+
                              ! Macro library statistics !
                              +-----------------------------+


Macro library name                          Macros defined
------------------                          --------------

_$255$DUA28:[SYSLOA.OBJ]CLUSTER.MLB;1              1
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                     2
_$255$DUA28:[SYSLIB]STARLET.MLB;2                  4
TOTALS (all libraries)                            7
```

363 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:CSPCLIENT/OBJ=OBJ$:CSPCLIENT MSRC$:CSPCLIENT/UPDATE=(ENH$:CSPCLIENT)+EXECML$/LIB+LIB$:CLUSTER/LIB

CSPOPCOM
LIS

CSPWAIT
LIS

CSPRCPCAC
LIS

CSPCJFRES
LIS

CSPQUORUM
LIS

DISTRLKI
LIS

CSPMOUNT
LIS

CSPVECTOR
LIS

CSPCLIENT
LIS

DSTRLOCK
LIS

DSTRLOCK
LIS