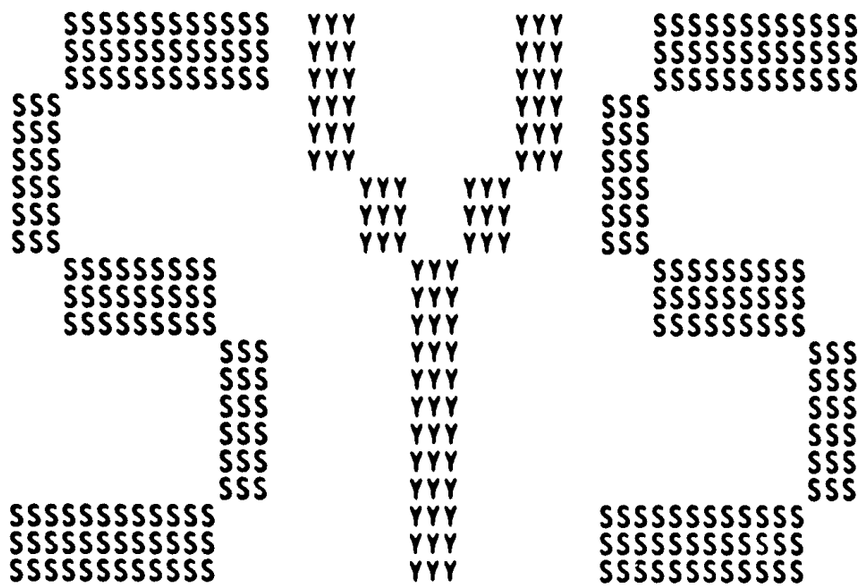

```
  CCCCCCCC  LL          UU      UU    SSSSSSSS  TTTTTTTTTT  RRRRRRRR    VV      VV  EEEEEEEEEE    CCCCCCCC
  CCCCCCCC  LL          UU      UU    SSSSSSSS  TTTTTTTTTT  RRRRRRRR    VV      VV  EEEEEEEEEE    CCCCCCCC
CC          LL          UU      UU  SS              TT      RR      RR  VV      VV  EE          CC
CC          LL          UU      UU  SS              TT      RR      RR  VV      VV  EE          CC
CC          LL          UU      UU  SS              TT      RR      RR  VV      VV  EE          CC
CC          LL          UU      UU  SS              TT      RR      RR  VV      VV  EE          CC
CC          LL          UU      UU    SSSSSS        TT      RRRRRRRR    VV      VV  EEEEEEEE    CC
CC          LL          UU      UU    SSSSSS        TT      RRRRRRRR    VV      VV  EEEEEEEE    CC
CC          LL          UU      UU          SS      TT      RR  RR      VV      VV  EE          CC
CC          LL          UU      UU          SS      TT      RR  RR      VV      VV  EE          CC
CC          LL          UU      UU          SS      TT      RR    RR      VV  VV    EE          CC                ....
CC          LL          UU      UU          SS      TT      RR    RR      VV  VV    EE          CC                ....
  CCCCCCCC  LLLLLLLLLL  UUUUUUUUUU  SSSSSSSS        TT      RR      RR      VV      EEEEEEEEEE    CCCCCCCC      ....
  CCCCCCCC  LLLLLLLLLL  UUUUUUUUUU  SSSSSSSS        TT      RR      RR      VV      EEEEEEEEEE    CCCCCCCC      ....

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
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .IF     NDF,PRMSW
0000     2          .TITLE  CLUSTRVEC - Cluster Loadable Code Vectors
0000     3          .IFF
0000     4          .TITLE  CLUSTRLOA - Cluster Loadable Code
0000     5          .ENDC
0000     6          .IDENT  'V04-000'
0000     7
0000     8 ;
0000     9 ;*****************************************************************************
0000    10 ;*                                                                           *
0000    11 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    12 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    13 ;*  ALL RIGHTS RESERVED.                                                     *
0000    14 ;*                                                                           *
0000    15 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    16 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    17 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    18 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    19 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    20 ;*  TRANSFERRED.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    23 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    24 ;*  CORPORATION.                                                             *
0000    25 ;*                                                                           *
0000    26 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    27 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    28 ;*                                                                           *
0000    29 ;*                                                                           *
0000    30 ;*****************************************************************************
0000    31 ;
0000    32
0000    33 ;++
0000    34 ;
0000    35 ; FACILITY:     Executive, system services and fork level code
0000    36 ;
0000    37 ; ABSTRACT:
0000    38 ;       This module contains the entry point vectors used to implement
0000    39 ;       VMS cluster functions as well as the table used to hook up those
0000    40 ;       vectors to the actual routines.  This module is assembled in two
0000    41 ;       different ways.  If PRMSW is defined the resultant module (CLUSTRLOA)
0000    42 ;       is linked with the actual loadable routines.  If PRMSW is undefined,
0000    43 ;       the resultant module (CLUSTRVEC) is linked with SYS.EXE.
0000    44 ;
0000    45 ; ENVIRONMENT:  Kernel mode, fork level
0000    46 ;
0000    47 ;--
0000    48 ;
0000    49 ; AUTHOR: Steve Beckhardt,       CREATION DATE: 6-Jan-1983
0000    50 ;
0000    51 ; MODIFIED BY:
0000    52 ;
0000    53 ;       V03-020 DWT0239         David W. Thiel          29-Aug-1984
0000    54 ;               Add CNX$BUGCHECK_CLUSTER entry point.
0000    55 ;
0000    56 ;       V03-019 DWT0207         David W. Thiel          09-Apr-1984
0000    57 ;               Add CNX$POWER_FAIL vector to be called on a
```

```
0000     58 ;               recovery from a power failure. Add CNX$DISK_CHANGE
0000     59 ;               so that SYSINIT and CSP can inform connection
0000     60 ;               manager when quorum disk is found.
0000     61 ;
0000     62 ;       V03-018 SRB0117         Steve Beckhardt         18-Mar-1984
0000     63 ;               Added vectors to get to routines for distributed
0000     64 ;               deadlock detection.
0000     65 ;
0000     66 ;       V03-017 DWT0177         David W. Thiel          27-Feb-1984
0000     67 ;               Remove CNX$DEALL_WARMCDRP and CNX$DEALL_MSG_BUF.
0000     68 ;               Add CNX$SHUTDOWN entry point.  Rearrange code to
0000     69 ;               be more conservative about deallocating code under
0000     70 ;               the PC.  Change the trailer PSECT name to be later
0000     71 ;               in the collating sequence.  Add patch space.
0000     72 ;
0000     73 ;       V03-016 ADE0001         Alan D. Eldridge        10-Jan-1983
0000     74 ;               Add EXE$CSP_BRDCST to vectors.
0000     75 ;
0000     76 ;       V03-015 ADE0001         Alan D. Eldridge        10-Jan-1983
0000     77 ;               Add EXE$CSP_COMMAND to vectors and call to CSP$INIT in
0000     78 ;               initialization call.
0000     79 ;
0000     80 ;       V03-014 RSH0051         R. Scott Hanna          11-AUG-1983
0000     81 ;               Add call to CNX$QUORUM_INIT
0000     82 ;
0000     83 ;       V03-013 ROW0197         Ralph O. Weber          29-JUL-1983
0000     84 ;               Add LOADVEC definitions for CNX$PARTNER_RESPOND, a respond to
0000     85 ;               block transfer service, and CNX$CHANGE_QUORUM, a hook for a
0000     86 ;               future change QUORUM dynamically service.
0000     87 ;
0000     88 ;       V03-012 ROW0185         Ralph O. Weber          24-JUN-1983
0000     89 ;               Add yet another ton of LOADVEC statements to define entries
0000     90 ;               for acknowledged message block transfer services.
0000     91 ;
0000     92 ;       V03-011 RNG0011         Rod N. Gamache          13-Jun-1983
0000     93 ;               Add entry points for the distributed GETLKI system service.
0000     94 ;
0000     95 ;       V03-010 JLV0268         Jake VanNoy             27-MAY-1983
0000     96 ;               Add EXE$CSP_BRKTHRU, entry point for cluster broadcasts.
0000     97 ;
0000     98 ;       V03-009 ROW0179         Ralph O. Weber          29-APR-1983
0000     99 ;               Add numerous LOADVEC statements to define entries for new
0000    100 ;               acknowledged message services and distributed lock manager
0000    101 ;               support.
0000    102 ;
0000    103 ;       V03-008 PRB0162         Paul R. Beck            14-APR-1983
0000    104 ;               Add vectors for EXE$ALLOC_CSD, EXE$DEALLOC_CSD, EXE$CSP_CALL.
0000    105 ;
0000    106 ;       V03-007 JWH0212         Jeffrey W. Horn         13-Apr-1983
0000    107 ;               Fix mistype in JWH0207.
0000    108 ;
0000    109 ;       V03-006 JWH0207         Jeffrey W. Horn         12-Apr-1983
0000    110 ;               Use SLVTAB macro to generate load-code prologue.
0000    111 ;
0000    112 ;       V03-005 ROW0174         Ralph O. Weber          29-MAR-1983
0000    113 ;               Add LOADVEC definitions for CNX$ALLOC_CDRP, CNX$SEND_MSG,
0000    114 ;               CNX$SEND_MSG_RESP, and CNX$SEND_MSG_RSPID; all routines which
```

```
0000    115 ;               the common journaling facility must call to perform cluster
0000    116 ;               journaling.
0000    117 ;
0000    118 ;       V03-004 DWT0087         David W. Thiel          23-Mar-1983
0000    119 ;               Correct previous modification to return valid
0000    120 ;               status after initialization.
0000    121 ;
0000    122 ;       V03-003 SRB0069         Steve Beckhardt         11-Mar-1983
0000    123 ;               Added support to allow SYSENQDEQ to be linked with
0000    124 ;               CLUSTRLOA for debugging purposes.  Removed cell
0000    125 ;               LCK$GL_RQSEQNM (it is now in SYSCOMMON).
0000    126 ;
0000    127 ;       V03-002 DWT0070         David W. Thiel          28-Jan-1983
0000    128 ;               Setup to initialize automatically on being loaded.
0000    129 ;               Adjust psects and alignment.
0000    130 ;
```

```
                                        .SBTTL  DECLARATIONS
                                ;
                                ; INCLUDE FILES:
                                ;

                                ;
                                ; MACROS:
                                ;

                                        .IF     DF,PRMSW
                                        $DYNDEF                         ; Data structure names
                                        .IF_TRUE_FALSE
                                        $SLVDEF                         ; Loadable vector definitions

                                ;
                                ; EQUATED SYMBOLS:
                                ;

                                ;
                                ; VECTOR LIST
                                ;

                                        .IF_TRUE                        ; For inclusion with loadable code

                                        .PSECT  ___999,4                ; Octaword alignment

                                PATCH_BEGIN:
                                        .BLKB   256                     ; Patch space
                                PATCH_END:

                                CLULOA_END:                             ; End of loadable code

                                        .PSECT  $$$000,4

                                CLULOA_START:
                                        SLVTAB  END     =       CLULOA_END, -
                                                INITRTN =       CLULOA_INIT, -
                                                SUBTYP  =       DYN$C_CC_CLS, -
                                                FACILITY=       <CLUSTER>

                                        .IF_FALSE                       ; For inclusion with SYS.EXE
                                        .PSECT  $$$500,LONG

                                        .ALIGN  LONG
                                CLU$GL_CLUB::                           ; Address of CLUster Block
00000000                        .LONG   0
                                CLU$GL_CLUSVEC::                        ; Address of Cluster System vector
00000000                        .LONG   0
                                CLU$GW_MAXINDEX::                       ; Maximum index+1 in Cluster System vector
0000                            .WORD   0
                                CLU_RSB:                                ; Used to make unloaded entry a NOP
05                              RSB

                                        .ALIGN  LONG
                                CLU$AL_LOAVEC::                         ; New symbol to replace next
                                CLS$AL_LOAVEC::
                                        .IF_TRUE_FALSE
```

```
                        189
000C    000C    190             LOADVEC CLU$GL_LOA_ADDR,1,,CLULOA_START ; Cluster code load address
0010    0010    191             LOADVEC LCK$SND_CVTREQ
0016    0016    192             LOADVEC LCK$SND_LOCKREQ
001C    001C    193             LOADVEC LCK$SND_GRANTED
0022    0022    194             LOADVEC LCK$SND_DEQGR
0028    0028    195             LOADVEC LCK$SND_DEQCV
002E    002E    196             LOADVEC LCK$SND_DEQWT
0034    0034    197             LOADVEC LCK$SND_BLKING
003A    003A    198             LOADVEC LCK$SND_RMVDIR
0040    0040    199             LOADVEC LCK$SND_TIMESTAMP_RQST
0046    0046    200             LOADVEC LCK$SND_SRCHDLCK
004C    004C    201             LOADVEC LCK$SND_DLCKFND
0052    0052    202             LOADVEC LCK$SND_REDO_SRCH
0058    0058    203             LOADVEC LCK$CVT_ID_TO_LKB
005E    005E    204             LOADVEC CNX$ALLOC_CDRP
0064    0064    205             LOADVEC CNX$ALLOC_CDRP_ONLY
006A    006A    206             LOADVEC CNX$ALLOC_WARMCDRP
0070    0070    207             LOADVEC CNX$ALLOC_WARMCDRP_CSB
0076    0076    208             LOADVEC CNX$DEALL_MSG_BUF_CSB
007C    007C    209             LOADVEC CNX$DEALL_WARMCDRP_CSB
0082    0082    210             LOADVEC CNX$INIT_CDRP
0088    0088    211             LOADVEC CNX$SEND_MNY_MSGS
008E    008E    212             LOADVEC CNX$SEND_MSG
0094    0094    213             LOADVEC CNX$SEND_MSG_CSB
009A    009A    214             LOADVEC CNX$SEND_MSG_RESP
00A0    00A0    215             LOADVEC CNX$SEND_MSG_RSPID
00A6    00A6    216             LOADVEC CNX$BLOCK_XFER
00AC    00AC    217             LOADVEC CNX$BLOCK_XFER_IRP
00B2    00B2    218             LOADVEC CNX$PARTNER_INIT_CSB
00B8    00B8    219             LOADVEC CNX$BLOCK_READ
00BE    00BE    220             LOADVEC CNX$BLOCK_READ_IRP
00C4    00C4    221             LOADVEC CNX$BLOCK_WRITE
00CA    00CA    222             LOADVEC CNX$BLOCK_WRITE_IRP
00D0    00D0    223             LOADVEC CNX$PARTNER_FINISH
00D6    00D6    224             LOADVEC CNX$PARTNER_RESPOND
00DC    00DC    225             LOADVEC CNX$CHANGE_QUORUM
00E2    00E2    226             LOADVEC CNX$SHUTDOWN
00E8    00E8    227             LOADVEC CNX$POWER_FAIL,,CLU_RSB
00EE    00EE    228             LOADVEC CNX$DISK_CHANGE
00F4    00F4    229             LOADVEC CNX$BUGCHECK_CLUSTER
00FA    00FA    230             LOADVEC EXE$ALLOC_CSD
0100    0100    231             LOADVEC EXE$DEALLOC_CSD
0106    0106    232             LOADVEC EXE$CSP_BRDCST
010C    010C    233             LOADVEC EXE$CSP_CALL
0112    0112    234             LOADVEC EXE$CSP_COMMAND
0118    0118    235             LOADVEC EXE$CSP_BRKTHRU
011E    011E    236             LOADVEC LKI$SND_STDREQ
0124    0124    237             LOADVEC LKI$SND_BLKING
012A    012A    238             LOADVEC LKI$SND_BLKBY
0130    0130    239             LOADVEC LKI$SND_LOCKS
0136    0136    240
0136    0136    241             .IF_TRUE                                ; End of vector list
0136    0136    242             .LONG   -1
0136    0136    243
0136    0136    244 CLULOA_INIT:                                        ; Initialization code
0136    0136    245             BSBW    CNX$INIT                        ; Do initializtion
```

```
0136   246          BSBW    CNX$QUORUM_INIT         ; Perform quorum file initialization
0136   247          BLBC    R0,20$                  ; Branch on error
0136   248          BSBW    CSP$INIT                ; Do load CSP code initialization
0136   249
0136   250          ; If the $ENQ and $DEQ system services (module SYSENQDEQ) are linked
0136   251          ; with this (for debugging purposes) then hook up the system service
0136   252          ; vectors to point these routines.
0136   253          ; NOTE:  The SYSGEN parameter SYSPAGING MUST be set to 0 for this
0136   254          ; to work.
0136   255
0136   256          .WEAK   EXE$$ENQ,EXE$$DEQ
0136   257
0136   258          TSTL    #EXE$$ENQ               ; Is loadable $ENQ service present?
0136   259          BEQL    10$                     ; Not linked with this module
0136   260          JSB     G^INI$WRITABLE          ; Make system writeable
0136   261          MOVW    B^30$,G^EXE$ENQ+2       ; Move JMP @# to start of resident
0136   262          MOVW    B^30$,G^EXE$DEQ+2       ; services
0136   263          MOVAB   W^EXE$$ENQ+2,G^EXE$ENQ+4; Move addresses of loadable services
0136   264          MOVAB   W^EXF$$DEQ+2,G^EXE$DEQ+4; to complete JMP @# instructions
0136   265          JSB     G^INI$RDONLY            ; Make system read only again
0136   266
0136   267 10$:     MOVAB   G^EXE$GL_NONPAGED+4,R3  ; Address of non-paged pool listhead
0136   268          MOVAB   W^CLULOA_START,R0       ; Address of block to deallocate
0136   269          MOVZWL  #CLULOA_INIEND-CLULOA_START,R1  ; Length of block to deallocate
0136   270          SUBW3   R1,8(R0),W^CLULOA_INIEND+8      ; Compute new length
0136   271          BRW     DEALLOCATE              ; Deallocate this piece of memory
0136   272
0136   273 20$:     RSB                             ; Return
0136   274
0136   275 30$:     JMP     G^EXE$ENQ               ; Absolute jump instruction
0136   276
0136   277 ; Descriptor for patch space available in this module
0136   278 ; Note that this space is deallocated when the code is loaded
0136   279 ;
0136   280 CNX$PATCH::
0136   281          .WORD   PATCH_END-PATCH_BEGIN
0136   282          .WORD   0
0136   283          .LONG   PATCH_BEGIN
0136   284
0136   285          .PSECT  $$$010,4                ; Octaword alignment
0136   286
0136   287 ; When initialization is complete and the initialization code is
0136   288 ; deleted, this block is left at the head of the remaining code.
0136   289 ;
0136   290          .ALIGN  4
0136   291 CLULOA_INIEND:                           ; End of initialization code
0136   292          SLVTAB  END     =       CLULOA_END, -
0136   293                  SUBTYP  =       DYN$C_CC_CLS, -
0136   294                  FACILITY=       <CLUSTERS>
0136   295
0136   296
0136   297 DEALLOCATE:
0136   298          JSB     G^EXE$DEALLOCATE        ; Free initialization code
0136   299          MOVZWL  S^#SS$_NORMAL,R0        ; Set success status
0136   300          RSB                             ; Return from initialization
0136   301
0136   302          .ENDC
```

```
                0136    303
                0136    304             .END
```

CLS$AL_LOAVEC	0000000C	RG	02
CLU$AL_LOAVEC	0000000C	RG	02
CLU$GL_CLUB	00000000	RG	02
CLU$GL_CLUSVEC	00000004	RG	02
CLU$GL_LOA_ADDP	0000000C	RG	02
CLU$GW_MAXINDEX	00000008	RG	02
CLU_RSB	0000000A	R	02
CNX$ALLOC_CDRP	0000005E	RG	02
CNX$ALLOC_CDRP_ONLY	00000064	RG	02
CNX$ALLOC_WARMCDRP	0000006A	RG	02
CNX$ALLOC_WARMCDRP_CSB	00000070	RG	02
CNX$BLOCK_READ	000000B8	RG	02
CNX$BLOCK_READ_IRP	000000BE	RG	02
CNX$BLOCK_WRITE	000000C4	RG	02
CNX$BLOCK_WRITE_IRP	000000CA	RG	02
CNX$BLOCK_XFER	000000A6	RG	02
CNX$BLOCK_XFER_IRP	000000AC	RG	02
CNX$BUGCHECK_CLUSTER	000000F4	RG	02
CNX$CHANGE_QUORUM	000000DC	RG	02
CNX$DEALL_MSG_BUF_CSB	00000076	RG	02
CNX$DEALL_WARMCDRP_CSB	0000007C	RG	02
CNX$DISK_CHANGE	000000EE	RG	02
CNX$INIT_CDRP	00000082	RG	02
CNX$PARTNER_FINISH	000000D0	RG	02
CNX$PARTNER_INIT_CSB	000000B2	RG	02
CNX$PARTNER_RESPOND	000000D6	RG	02
CNX$POWER_FAIL	000000E8	RG	02
CNX$SEND_ANY_MSGS	00000088	RG	02
CNX$SEND_MSG	0000008E	RG	02
CNX$SEND_MSG_CSB	00000094	RG	02
CNX$SEND_MSG_RESP	0000009A	RG	02
CNX$SEND_MSG_RSPID	000000A0	RG	02
CNX$SHUTDOWN	000000E2	RG	02
EXE$ALLOC_CSD	000000FA	RG	02
EXE$CSP_BRDCST	00000106	RG	02
EXE$CSP_BRKTHRU	00000118	RG	02
EXE$CSP_CALL	0000010C	RG	02
EXE$CSP_COMMAND	00000112	RG	02
EXE$DEALLOC_CSD	00000100	RG	02
EXE$LOAD_ERROR	********	X	02
LCK$CVT_ID_TO_LKB	00000058	RG	02
LCK$SND_BLKING	00000034	RG	02
LCK$SND_CVTREQ	00000010	RG	02
LCK$SND_DEQCV	00000028	RG	02
LCK$SND_DEQGR	00000022	RG	02
LCK$SND_DEQWT	0000002E	RG	02
LCK$SND_DLCKFND	0000004C	RG	02
LCK$SND_GRANTED	0000001C	RG	02
LCK$SND_LOCKREQ	00000016	RG	02
LCK$SND_REDO_SRCH	00000052	RG	02
LCK$SND_RMVDIR	0000003A	RG	02
LCK$SND_SRCHDLCK	00000046	RG	02
LCK$SND_TIMESTAMP_RQST	00000040	RG	02
LKI$SND_BLKBY	0000012A	RG	02
LKI$SND_BLKING	00000124	RG	02
LKI$SND_LOCKS	00000130	RG	02
LKI$SND_STDREQ	0000011E	RG	02

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
$$$500	00000136 (310.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:01.07
Command processing	130	00:00:00.54	00:00:03.79
Pass 1	159	00:00:04.38	00:00:11.68
Symbol table sort	0	00:00:00.06	00:00:00.09
Pass 2	72	00:00:01.24	00:00:03.54
Symbol table output	8	00:00:00.08	00:00:00.10
Psect synopsis output	1	00:00:00.02	00:00:00.15
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	407	00:00:06.39	00:00:20.43

The working set limit was 1200 pages.
20019 bytes (40 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 82 non-local and 0 local symbols.
304 source lines were read in Pass 1, producing 18 object records in Pass 2.
10 pages of virtual memory were used to define 8 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	2
_255DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	5

142 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:CLUSTRVEC/OBJ=OBJ$:CLUSTRVEC MSRC$:CLUSTR/UPDATE=(ENH$:CLUSTR)+EXECML$/LIB
```

0373 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CMODSSDSP
LIS

BUGCHECK
LIS

COMDRVSUB
LIS

CLUSTRVEC
LIS

BUFFERCTL
LIS

DEADLOCK
LIS

CVTFILNAM
LIS

BOOPARAM
LIS

CJFSYSVEC
LIS

CVTATB
LIS

BUGCHKMSG
LIS

CONSOLIO
LIS