



```

CCCCCCCC LL      UU      UU      SSSSSSSS TTTTTTTTTT RRRRRRRR LL      000000      AAAAAA
CCCCCCCC LL      UU      UU      SSSSSSSS TTTTTTTTTT RRRRRRRR LL      000000      AAAAAA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CC        LL      UU      UU      SS        TT        RR      RR      LL      00      00      AA      AA
CCCCCCCC LLLLLLLLLL UUUUUUUUUU SSSSSSSS TT        RR      RR      LLLLLLLLLL 000000      AA      AA
CCCCCCCC LLLLLLLLLL UUUUUUUUUU SSSSSSSS TT        RR      RR      LLLLLLLLLL 000000      AA      AA

```

```

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

```

(2) 132

DECLARATIONS

```

00000001 0000 1 PRMSW=1 ; SET SWITCH TO GENERATE PARAMETER DESCRIPTO
          0000 1 .IF NDF,PRMSW
          0000 2 .TITLE CLUSTERVEC - Cluster Loadable Code Vectors
          0000 3 .IFF
          0000 4 .TITLE CLUSTERLOA - 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 (CLUSTERLOA)
          0000 42 is linked with the actual loadable routines. If PRMSW is undefined,
          0000 43 the resultant module (CLUSTERVEC) 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 SI.VTAB 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	:	
0000	115	:	
0000	116	:	
0000	117	:	
0000	118	:	
0000	119	:	
0000	120	:	
0000	121	:	
0000	122	:	
0000	123	:	
0000	124	:	
0000	125	:	
0000	126	:	
0000	127	:	
0000	128	:	
0000	129	:	
0000	130	:	

CNX\$SEND\_MSG\_RESP, and CNX\$SEND\_MSG\_RSPID; all routines which  
the common journaling facility must call to perform cluster  
journaling.

V03-004 DWT0087 David W. Thiel 23-Mar-1983  
Correct previous modification to return valid  
status after initialization.

V03-003 SRB0069 Steve Beckhardt 11-Mar-1983  
Added support to allow SYSENGDEQ to be linked with  
CLUSTERLOA for debugging purposes. Removed cell  
LCK\$GL\_RQSEQNM (it is now in SYSCOMMON).

V03-002 DWT0070 David W. Thiel 28-Jan-1983  
Setup to initialize automatically on being loaded.  
Adjust psects and alignment.

```

0000 132      .SBTTL  DECLARATIONS
0000 133      :
0000 134      : INCLUDE FILES:
0000 135      :
0000 136      :
0000 137      :
0000 138      : MACROS:
0000 139      :
0000 140      :
0000 141      .IF      DF,PRMSW
0000 142      $DYNDDEF      ; Data structure names
0000 143      .IF TRUE_FALSE
0000 144      $SLVDEF      ; Loadable vector definitions
0000 145      :
0000 146      :
0000 147      : EQUATED SYMBOLS:
0000 148      :
0000 149      :
0000 150      :
0000 151      : VECTOR LIST
0000 152      :
0000 153      :
0000 154      .IF_TRUE      ; For inclusion with loadable code
0000 155      :
00000000 156      .PSECT  ___999,4      ; Octaword alignment
0000 157      :
00000100 0000 158 PATCH_BEGIN:
00000100 0000 159      .BLKB  256      ; Patch space
0100 160 PATCH_END:
0100 161 :
0100 162 CLULOA_END:      ; End of loadable code
0100 163 :
00000000 164      .PSECT  $$$000,4
0000 165 :
0000 166 CLULOA_START:
0000 167      SLVTAB  END      =      CLULOA_END, -
0000 168      INITRTN =      CLULOA_INIT, -
0000 169      SUBTYP  =      DYN$C [C CLS, -
0000 170      FACILITY=      <CLUSTERS>
0024 171 :
0024 172      .IF FALSE
0024 173      .PSECT  $$$500, LONG      ; For inclusion with SYS.EXE
0024 174 :
0024 175      .ALIGN  LONG
0024 176 CLUSGL_CLUB::      ; Address of CLUster Block
0024 177      .LONG  0
0024 178 CLUSGL_CLUSVEC::      ; Address of Cluster System vector
0024 179      .LONG  0
0024 180 CLUSGW_MAXINDEX::      ; Maximum index+1 in Cluster System vector
0024 181      .WORD  0
0024 182 CLU_RSB:      ; Used to make unloaded entry a NOP
0024 183      RSB
0024 184 :
0024 185      .ALIGN  LONG
0024 186 CLUSAL_LOAVEC::      ; New symbol to repiace next
0024 187 CLSSAL_LOAVEC::
0024 188      .IF TRUE_FALSE

```

```

0024 189
0024 190 LOADVEC CLUSGL LOA ADDR,1,,CLULOA_START ; Cluster code load address
0029 191 LOADVEC LCKSSND_CVTREQ
002E 192 LOADVEC LCKSSND_LOCKREQ
0033 193 LOADVEC LCKSSND_GRANTED
0038 194 LOADVEC LCKSSND_DEQGR
003D 195 LOADVEC LCKSSND_DEQCV
0042 196 LOADVEC LCKSSND_DEQWT
0047 197 LOADVEC LCKSSND_BLKING
004C 198 LOADVEC LCKSSND_RMVDIR
0051 199 LOADVEC LCKSSND_TIMESTAMP_RST
0056 200 LOADVEC LCKSSND_SRCHDLCK
005B 201 LOADVEC LCKSSND_DLCKFND
0060 202 LOADVEC LCKSSND_REDO_SRCH
0065 203 LOADVEC LCKSCVT_ID_TO_LKB
006A 204 LOADVEC CNXSALLOC_CDRP
006F 205 LOADVEC CNXSALLOC_CDRP_ONLY
0074 206 LOADVEC CNXSALLOC_WARMCDRP
0079 207 LOADVEC CNXSALLOC_WARMCDRP_CSB
007E 208 LOADVEC CNXSDEALL_MSG_BUF_CSB
0083 209 LOADVEC CNXSDEALL_WARMCDRP_CSB
0088 210 LOADVEC CNXSINIT_CDRP
008D 211 LOADVEC CNXSSEND_MNY_MSGS
0092 212 LOADVEC CNXSSEND_MSG
0097 213 LOADVEC CNXSSEND_MSG_CSB
009C 214 LOADVEC CNXSSEND_MSG_RESP
00A1 215 LOADVEC CNXSSEND_MSG_RSPID
00A6 216 LOADVEC CNXSBLOCK_XFER
00AB 217 LOADVEC CNXSBLOCK_XFER_IRP
00B0 218 LOADVEC CNXSPARTNER_INIT_CSB
00B5 219 LOADVEC CNXSBLOCK_READ
00BA 220 LOADVEC CNXSBLOCK_READ_IRP
00BF 221 LOADVEC CNXSBLOCK_WRITE
00C4 222 LOADVEC CNXSBLOCK_WRITE_IRP
00C9 223 LOADVEC CNXSPARTNER_FINISH
00CE 224 LOADVEC CNXSPARTNER_RESPOND
00D3 225 LOADVEC CNXSCHANGE_QUORUM
00D8 226 LOADVEC CNXSSHUTDOWN
00DD 227 LOADVEC CNX$POWER_FAIL,,CLU_RSB
00E2 228 LOADVEC CNX$DISK_CHANGE
00E7 229 LOADVEC CNX$BUGCHECK_CLUSTER
00EC 230 LOADVEC EXESALLOC_CSD
00F1 231 LOADVEC EXESDEALLOC_CSD
00F6 232 LOADVEC EXESCSP_BRDST
00FB 233 LOADVEC EXESCSP_CALL
0100 234 LOADVEC EXESCSP_COMMAND
0105 235 LOADVEC EXESCSP_BRKTHRU
010A 236 LOADVEC LKISSND_STDREQ
010F 237 LOADVEC LKISSND_BLKING
0114 238 LOADVEC LKISSND_BLKBY
0119 239 LOADVEC LKISSND_LOCKS
011E 240
011E 241 .IF TRUE ; End of vector list
FFFFFFF 011E 242 .LONG -1
0122 243
0122 244 CLULOA_INIT: ; Initialization code
FEDB' 30 0122 245 BSBW CNXSINIT ; Do initializtion

```



```

FED8' 30 0125 246 BSBW CNX$QUORUM_INIT ; Perform quorum file initialization
54 50 E9 0128 247 BLBC RO,20$ ; Branch on error
FED2' 30 012B 248 BSBW CSP$INIT ; Do load CSP code initialization
012E 249
012E 250 ; If the $ENQ and $DEQ system services (module SYSENQDEQ) are linked
012E 251 ; with this (for debugging purposes) then hook up the system service
012E 252 ; vectors to point these routines.
012E 253 ; NOTE: The SYSGEN parameter SYSPAGING MUST be set to 0 for this
012E 254 ; to work.
012E 255
012E 256 .WEAK EXES$ENQ,EXES$DEQ
012E 257
0000000'8F D5 012E 258 TSTL #EXES$ENQ ; Is loadable $ENQ service present?
2E 13 0134 259 BEQL 10$ ; Not linked with this module
0000000'GF 16 0136 260 JSB G^INIS$WRITABLE ; Make system writeable
0000002'GF 80'AF B0 013C 261 MOVW B^30$,G^EXE$ENQ+2 ; Move JMP @# to start of resident
0000002'GF 80'AF B0 0144 262 MOVW B^30$,G^EXE$DEQ+2 ; services
0000004'GF 0002'CF 9E 014C 263 MOVAB W^EXE$ENQ+2,G^EXE$ENQ+4 ; Move addresses of loadable services
0000004'GF 0002'CF 9E 0155 264 MOVAB W^EXE$DEQ+2,G^EXE$DEQ+4 ; to complete JMP @# instructions
0000000'GF 16 015E 265 JSB G^INIS$RDONLY ; Make system read only again
0164 266
53 0000004'GF 9E 0164 267 10$: MOVAB G^EXE$GL_NONPAGED+4,R3 ; Address of non-paged pool listhead
50 FE91 CF 9E 016B 268 MOVAB W^CLULOA_START,R0 ; Address of block to deallocate
51 0000'8F 3C 0170 269 MOVZWL #CLULOA_INIEND-CLULOA_START,R1 ; Length of block to deallocate
0008'CF 08 A0 51 A3 0175 270 SUBW3 R1,8(R0),W^CLULOA_INIEND+8 ; Compute new length
FEA5' 31 017C 271 BRW DEALLOCATE ; Deallocate this piece of memory
017F 272
05 017F 273 20$: RSB ; Return
0180 274
0000000'GF 17 0180 275 30$: JMP G^EXE$ENQ ; Absolute jump instruction
0186 276
0186 277 ; Descriptor for patch space available in this module
0186 278 ; Note that this space is deallocated when the code is loaded
0186 279
0186 280 CNX$PATCH::
0100 0186 281 .WORD PATCH_END-PATCH_BEGIN
0000 0188 282 .WORD 0
0000000C' 018A 283 .LONG PATCH_BEGIN
018E 284
00000000 285 .PSECT $$$010,4 ; Octaword alignment
0000 286
0000 287 ; When initialization is complete and the initialization code is
0000 288 ; deleted, this block is left at the head of the remaining code.
0000 289
0000 290 .ALIGN 4
0000 291 CLULOA_INIEND: ; End of initialization code
0000 292 SLVTAB END = CLULOA_END, -
0000 293 SUBTYP = DYN$C [C CLS, -
0000 294 FACILITY= <CLUSTERS>
0024 295
0024 296
0024 297 DEALLOCATE:
0000000'GF 16 0024 298 JSB G^EXE$DEALLOCATE ; Free initialization code
50 00' 3C 002A 299 MOVZWL S^#SS$NORMAL,R0 ; Set success status
05 002D 300 RSB ; Return from initialization
002E 301
002E 302 .ENDC

```

CLUSTERLOA  
V04-000

- Cluster Loadable Code  
DECLARATIONS

N 7

16-SEP-1984 00:24:20 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:33 [SYS.SRC]CLUSTER.MAR;1

Page 7  
(2)

CN  
VO

002E 303  
002E 304 .END

CLUSTRLOA  
Symbol table

- Cluster Loadable Code

B 8

16-SEP-1984 00:24:20 VAX/VMS Macro V04-00  
5-SEP-1984 03:40:33 [SYS.SRC]CLUSTR.MAR;1

Page 8  
(2)

CNX  
V04

CLULOA_END	00000100	R	02	LCK\$SND_DLCKFND	*****	X	03
CLULOA_INIEND	00000000	R	04	LCK\$SND_GRANTED	*****	X	03
CLULOA_INIT	00000122	R	03	LCK\$SND_LOCKREQ	*****	X	03
CLULOA_START	00000000	R	03	LCK\$SND_REDO_SRCH	*****	X	03
CNX\$ALLOC_CDRP	*****	X	03	LCK\$SND_RMVDIR	*****	X	03
CNX\$ALLOC_CDRP_ONLY	*****	X	03	LCK\$SND_SRCHDLCK	*****	X	03
CNX\$ALLOC_WARMCDRP	*****	X	03	LCK\$SND_TIMESTAMP_RST	*****	X	03
CNX\$ALLOC_WARMCDRP_CSB	*****	X	03	LKISSND_BLKBY	*****	X	03
CNX\$BLOCK_READ	*****	X	03	LKISSND_BLKING	*****	X	03
CNX\$BLOCK_READ_IRP	*****	X	03	LKISSND_LOCKS	*****	X	03
CNX\$BLOCK_WRITE	*****	X	03	LKISSND_STDREQ	*****	X	03
CNX\$BLOCK_WRITE_IRP	*****	X	03	PATCH_BEGIN	00000000	R	02
CNX\$BLOCK_XFER	*****	X	03	PATCH_END	00000100	R	02
CNX\$BLOCK_XFER_IRP	*****	X	03	PRMSW	= 00000001		
CNX\$BUGCHECK_CLUSTER	*****	X	03	PRT\$C_ER	= 00000007		
CNX\$CHANGE_QUORUM	*****	X	03	PRT\$C_EW	= 00000005		
CNX\$DEALL_MSG_BUF_CSB	*****	X	03	SS\$_NORMAL	*****	X	04
CNX\$DEALL_WARMCDRP_CSB	*****	X	03				
CNX\$DISK_CHANGE	*****	X	03				
CNX\$INIT	*****	X	03				
CNX\$INIT_CDRP	*****	X	03				
CNX\$PARTNER_FINISH	*****	X	03				
CNX\$PARTNER_INIT_CSB	*****	X	03				
CNX\$PARTNER_RESPOND	*****	X	03				
CNX\$PATCH	00000186	RG	03				
CNX\$POWER_FAIL	*****	X	03				
CNX\$QUORUM_INIT	*****	X	03				
CNX\$SEND_MNY_MSGS	*****	X	03				
CNX\$SEND_MSG	*****	X	03				
CNX\$SEND_MSG_CSB	*****	X	03				
CNX\$SEND_MSG_RESP	*****	X	03				
CNX\$SEND_MSG_RSPID	*****	X	03				
CNX\$SHUTDOWN	*****	X	03				
CSP\$INIT	*****	X	03				
DEALLOCATE	00000024	R	04				
DYN\$C_LC_CLS	= 00000005						
DYN\$C_LOADCODE	= 00000062						
EXE\$DEQ	*****	W	GX	03			
EXE\$ENQ	*****	W	GX	03			
EXE\$ALLOC_CSD	*****	X		03			
EXE\$CSP_BRDCST	*****	X		03			
EXE\$CSP_BRKTHRU	*****	X		03			
EXE\$CSP_CALL	*****	X		03			
EXE\$CSP_COMMAND	*****	X		03			
EXE\$DEALLOCATE	*****	X		04			
EXE\$DEALLOC_CSD	*****	X		03			
EXE\$DEQ	*****	X		03			
EXE\$ENQ	*****	X		03			
EXE\$GL_NONPAGED	*****	X		03			
IN:\$RDONLY	*****	X		03			
IN:\$WRITABLE	*****	X		03			
LCK\$CVT_ID_TO_LKB	*****	X		03			
LCK\$SND_BLKING	*****	X		03			
LCK\$SND_CVTREQ	*****	X		03			
LCK\$SND_DEQCV	*****	X		03			
LCK\$SND_DEQGR	*****	X		03			
LCK\$SND_DEQWT	*****	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
\$AB\$\$	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
999	00000100 ( 256.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC 20
\$\$\$000	0000018E ( 398.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC 20
\$\$\$010	0000002E ( 46.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC 20

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.06	00:00:00.51
Command processing	147	00:00:00.50	00:00:03.45
Pass 1	202	00:00:03.98	00:00:15.47
Symbol table sort	0	00:00:00.20	00:00:00.20
Pass 2	74	00:00:00.94	00:00:04.17
Symbol table output	9	00:00:00.06	00:00:00.29
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	473	00:00:05.76	00:00:24.11

The working set limit was 1350 pages.  
28211 bytes (56 pages) of virtual memory were used to buffer the intermediate code.  
There were 20 pages of symbol table space allocated to hold 257 non-local and 7 local symbols.  
305 source lines were read in Pass 1, producing 22 object records in Pass 2.  
12 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	0
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	9

394 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:CLUSTRLOA/OBJ=OBJ\$:CLUSTRLOA MASD\$:[SYS.SRC]PRMSW/UPDATE=(MASD\$:[SYS.ENH]PRMSW)+MASD\$:[SYS.SRC]CLUSTR/UPDATE=(MASD\$:[

