

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The letter is symmetrical and composed of multiple rows of these characters. The top of the 'A' is formed by a row of 'S's, followed by rows of 'Y's and 'S's. The sides of the 'A' are formed by rows of 'S's and 'Y's. The bottom of the 'A' is formed by a row of 'S's. The overall shape is a wide, triangular 'A' with a small gap in the center of the top bar.

```
LL      000000  AAAAAA  VV      VV  EEEEEEEEE  CCCCCCCC
LL      000000  AAAAAA  VV      VV  EEEEEEEEE  CCCCCCCC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EEEEEEE  CC
LL      00      00  AA      AA  VV      VV  EEEEEEE  CC
LL      00      00  AAAAAA  VV      VV  EE      CC
LL      00      00  AAAAAA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LL      00      00  AA      AA  VV      VV  EE      CC
LLLLLLLL 000000  AA      AA  VV      VV  EEEEEEEEE  CCCCCCCC
LLLLLLLL 000000  AA      AA  VV      VV  EEEEEEEEE  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
LLLLLLLL IIIIII  SSSSSSSS
LLLLLLLL IIIIII  SSSSSSSS
```

```
00000001 0000 1 PRMSW=1 ; SET SWITCH TO GENERATE PARAMETER DESCRIPTO
0000 1 .IF NDF,PRMSW ;FOR LINKAGE WITH SYS.EXE,...
0000 2 .TITLE SYSLOAVEC - SYSTEM VECTORS TO LOADABLE SUBROUTINES
0000 3
0000 4 .IFF ;FOR LINKAGE WITH SYSLOAxxx.EXE,...
0000 5 .TITLE LOAVEC - OFFSETS TO LOADABLE SUBROUTINES
0000 6 .ENDC
0000 7
0000 8 .IDENT 'V04-000'
0000 9
0000 10 :*****
0000 11 :*
0000 12 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 13 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 14 :* ALL RIGHTS RESERVED.
0000 15 :*
0000 16 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 17 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 18 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 19 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 20 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 21 :* TRANSFERRED.
0000 22 :*
0000 23 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 24 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 25 :* CORPORATION.
0000 26 :*
0000 27 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 28 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 29 :*
0000 30 :*
0000 31 :*****
0000 32 :
0000 33 :++
0000 34 :
0000 35 : FACILITY:
0000 36 :
0000 37 : EXECUTIVE, LOADABLE SUBROUTINES
0000 38 :
0000 39 : ABSTRACT:
0000 40 :
0000 41 : VECTORS, LOAD IMAGE SIZE, AND UNDEFINED VECTOR HANDLER FOR
0000 42 : LOADABLE SUBROUTINES.
0000 43 :
0000 44 : AUTHOR:
0000 45 :
0000 46 : N. KRONENBERG, MARCH 13, 1979.
0000 47 :
0000 48 : MODIFIED BY:
0000 49 :
0000 50 : V03-021 ROW0407 Ralph O. Weber 25-JUL-1984
0000 51 : Add EXESMNTVER_DVI_ASSIST, an escape transfer vector which
0000 52 : allows $GETDVI support for shadow sets to be shipped with
0000 53 : mount verification in SYSLOAxxx.
0000 54 :
0000 55 : V03-020 WHM0001 Bill Matthews 09-Jul-1984
0000 56 : Added vectors for routines CON$INIT_CTY, CON$GETCHAR and CON$PUTCHAR
```

0000	57	:	
0000	58	:	
0000	59	:	V03-019 ROW0384 Ralph O. Weber 7-JUL-1984
0000	60	:	Add the following transfer or pointer vectors for Mount
0000	61	:	Verification:
0000	62	:	o EXE\$CLUTRANIO, VAXcluster state transition I/O block.
0000	63	:	o EXE\$UPDGNERNUM, update shadow set generation number.
0000	64	:	o EXE\$GL_MVMSLBAS, MVMSL base address.
0000	65	:	
0000	66	:	V03-018 TCM0003 Trudy C. Matthews 09-Apr-1984
0000	67	:	Add vector for routine CON\$RELEASECTY.
0000	68	:	
0000	69	:	V03-017 ROW0330 Ralph O. Weber 24-MAR-1984
0000	70	:	Add two spare mount verification transfer vectors for possible
0000	71	:	use in volume shadowing.
0000	72	:	
0000	73	:	V03-016 KPL0001 Peter Lieberwirth 4-Mar-1984
0000	74	:	Add some extra vectors (aligned and packed) for possible
0000	75	:	use in 4.x.
0000	76	:	
0000	77	:	V03-015 ROW0317 Ralph O. Weber 27-FEB-1984
0000	78	:	Correct patch area descriptor to have patch size in bytes not
0000	79	:	in longwords.
0000	80	:	
0000	81	:	V03-014 ROW0292 Ralph O. Weber 4-FEB-1984
0000	82	:	Add three vectors for mount verification, which is being moved
0000	83	:	to SYSLOAxxx: EXE\$MOUNTVER the main entry point, EXE\$MNTVERSIO
0000	84	:	entry to the start I/O routine for mount verification, and
0000	85	:	EXE\$MNTVERSHDOL entry point to the bring shadow unit online
0000	86	:	routine (which does not exist yet). EXE\$MNTVERSIO allows
0000	87	:	in-driver mount verification routines to use a standard
0000	88	:	interface to starting and completing an internal I/O request.
0000	89	:	EXE\$MNTVERSHDOL is provided for MOUNT to use when it brings
0000	90	:	the members of a shadow set online asynchronously.
0000	91	:	
0000	92	:	V03-013 CWH8001 CW Hobbs 4-Dec-1983
0000	93	:	Add EXE\$READP_TODR and EXE\$WRITEP_TODR routines, used to
0000	94	:	force access of physical TODR on Nautilus CPU. On other
0000	95	:	CPUs, these routines are second labels on EXE\$READ_TODR and
0000	96	:	EXE\$WRITE_TODR.
0000	97	:	
0000	98	:	V03-012 KDM0062 Kathleen D. Morse 19-Jul-1983
0000	99	:	Add routine to initialize the time-wait macro data
0000	100	:	cells, EXE\$INI_TIMWAIT.
0000	101	:	
0000	102	:	V03-011 KTA0001 Kerbey T. Altmann 12-Jul-1983
0000	103	:	Add routine for console device data structure
0000	104	:	initialization, INI\$CONSOLE.
0000	105	:	
0000	106	:	V03-010 KDM0057 Kathleen D. Morse 12-Jul-1983
0000	107	:	Change the SYSINIT routine, SIP_SETTIME, to a loadable,
0000	108	:	cpu-dependent routine, EXE\$INIT_TODR.
0000	109	:	
0000	110	:	V03-009 KDM0048 Kathleen D. Morse 07-Jun-1983
0000	111	:	Add loadable routines for referencing TODR:
0000	112	:	EXE\$READ_TODR and EXE\$WRITE_TODR.
0000	113	:	
0000	113	:	V03-008 WMC0001 Wayne Cardoza 09-Jun-1983

0000	114	:	Add loadable code system service dispatchers.
0000	115	:	
0000	116	:	V03-007 JWH0203 Jeffrey W. Horn 22-Mar-1983
0000	117	:	Use SLVTAB macro to set up loadable-image header.
0000	118	:	
0000	119	:	V03-006 TCM0002 Trudy C. Matthews 16-Feb-1983
0000	120	:	Add CON\$OWNCTY vector.
0000	121	:	
0000	122	:	V03-005 TCM0001 Trudy C. Matthews 13-Jan-1983
0000	123	:	Add SYSL\$CLRSBIA entry point. Add an alternative default
0000	124	:	routine that RSBs harmlessly if called before code is
0000	125	:	loaded, instead of halting.
0000	126	:	
0000	127	:	V03-004 KTA3018 Kerbey T. Altmann 30-Oct-1982
0000	128	:	Redo to reflect changes in modules.
0000	129	:	
0000	130	:	V03-003 MSH0001 Maryann Hinden 24-Sep-1982
0000	131	:	Change entry EXE\$DW780_INT to EXE\$UBAERR_INT.
0000	132	:	
0000	133	:	V03-002 STJ3026 Steven T. Jeffreys 24-Sep-1982
0000	134	:	Moved LOADVEC macro to SYSMAR.
0000	135	:	
0000	136	:	V03-001 BLS0183 Benn Schreiber 16-Aug-1982
0000	137	:	Add new entry points for loadable console support
0000	138	:	
0000	139	:--	

```
0000 141 :+
0000 142 : THIS MODULE MAY BE ASSEMBLED WITH EXECUTIVE MODULE PRMSW.MAR
0000 143 : WHICH DEFINES THE CONDITIONAL CONTROL VARIABLE PRMSW. IF PRMSW
0000 144 : IS NOT DEFINED, THEN SYSLOAVEC.MAR IS INTENDED TO LINK WITH SYS.EXE.
0000 145 : IF PRMSW IS DEFINED, SYSLOAVEC.MAR IS INTENDED FOR LINKAGE WITH
0000 146 : SYSLOAxxx.EXE. (xxx IS THE CPU DESIGNATOR, E.G., 780.)
0000 147 :
0000 148 : IF PRMSW IS NOT DEFINED (LINK WITH SYS.EXE), THE SOURCE CONTAINS
0000 149 : A LIST OF VECTORS TO BE CONNECTED TO THE SUBROUTINES AND DATA STRUCTURES
0000 150 : IN SYSLOAxxx.EXE.
0000 151 :
0000 152 : DATA VECTORS ARE LONGWORD POINTERS TO THE LOADED DATA STRUCTURES. BEFORE
0000 153 : LOADING, THE POINTERS CONTAIN A 0.
0000 154 :
0000 155 : SUBROUTINE VECTORS ARE OF THE FORM:
0000 156 :
0000 157 : ENTRY:: JMP @#EXESLOAD_ERROR
0000 158 :
0000 159 : THERE ARE TWO TYPES OF ROUTINE ENTRIES. THE FIRST TYPE IS LONGWORD
0000 160 : ALIGNED ENTRIES FOR SCB VECTORS. THE SECOND TYPE IS PACKED
0000 161 : ENTRIES FOR ORDINARY ROUTINES.
0000 162 :
0000 163 : SYSLOAVEC ALSO CONTAINS THE LOAD ERROR HANDLER, EXESLOAD_ERROR,
0000 164 : WHICH IS SIMPLY A HALT. IF ANY LOADABLE SUBROUTINE IS CALLED
0000 165 : PRIOR TO BEING LOADED, THE HALT WILL BE EXECUTED.
0000 166 :
0000 167 : IF PRMSW IS DEFINED (LINK WITH SYSLOAxxx.EXE), THE SOURCE
0000 168 : CONTAINS LONGWORD EXESLOAD_SIZE, THE NUMBER OF BYTES IN SYSLOAxxx.EXE.
0000 169 : THE SOURCE ALSO CONTAINS A LIST OF THE SELF-RELATIVE OFFSETS TO
0000 170 : THE SUBROUTINES IN SYSLOAxxx.EXE. THE LIST OF OFFSETS IS USED
0000 171 : TO CONNECT THE SYS.EXE VECTORS TO THE LOADED SUBROUTINES.
0000 172 :-
```

```
0000 174 : MISC. LABELS AND DATA:
0000 175 :
0000 176 :
0000 177 :
0000 178 .IF DF,PRMSW ;FOR LINKAGE WITH SYSLOAxxx.EXE,...
0000 179
0000 180 $DYNDEF
0000 181
0000 182 .PSECT __LOAD_END,QUAD ; Force label to be at the end of
0000 183 SYSL$END:: ; loadable code
0000 184
0000 185 .PSECT $$$000 ; Force return to start of loadable code
0000 186 SYSL$BEGIN::
0000 187 SLVTAB END = SYSL$END, -
0000 188 INITRTN = INIS$IOMAP, -
0000 189 FACILITY= <SYSLOAVEC>
0024 190 :
0024 191 : NOTE: The modules, INIADPxxx, understand how to unload the vectors
0024 192 : for routines that are only called once during initialization
0024 193 : of the system. Therefore, if the format of these vectors
0024 194 : change, then the code in INIADPxxx must also change.
0024 195 :
0024 196 :
0024 197 :
0024 198 .IFF ; For linkage with SYS.EXE,...
0024 199 .PSECT $$$500, LONG
0024 200 .ALIGN LONG ; Start vector list on longwd
0024 201 EXE$AL_LOAVEC:: ; Addr of start of vectors
0024 202 .ENDC
0024 203
0024 204 :
0024 205 : VECTOR LIST:
0024 206 : Define longword-aligned routine vectors.
0024 207 :
0024 208 :
0024 209 :
0024 210 LOADVEC EXE$MCHK,TYPE=2 ;MACHINE CHECK HANDLER
0029 211 LOADVEC EXE$INT54,TYPE=2 ;INTERRUPT, SCB VECTOR=^X54
002E 212 LOADVEC EXE$INT58,TYPE=2 ;INTERRUPT, SCB VECTOR=^X58
0033 213 LOADVEC EXE$INT5C,TYPE=2 ;INTERRUPT, SCB VECTOR=^X5C
0038 214 LOADVEC EXE$INT60,TYPE=2 ;INTERRUPT, SCB VECTOR=^X60
003D 215 LOADVEC UBA$UNIXINT,TYPE=2, - ;INTERRUPT, UNIBUS ITSELF
003D 216 SEC_LABEL=UBA_UNEXINT
0042 217 :
0042 218 : Extra aligned vectors. Current target of these vectors in SYSLOA is a
0042 219 : halt instruction in ERRSUB*.
0042 220 :
0042 221 LOADVEC EXE$EXTRA1,TYPE=2 ;EXTRA LABEL
0047 222 LOADVEC EXE$EXTRA2,TYPE=2 ;EXTRA LABEL
004C 223 LOADVEC EXE$EXTRA3,TYPE=2 ;EXTRA LABEL
0051 224 LOADVEC EXE$EXTRA4,TYPE=2 ;EXTRA LABEL
0056 225 LOADVEC EXE$EXTRA5,TYPE=2 ;EXTRA LABEL
005B 226 :
005B 227 : Define packed routine vectors.
005B 228 :
005B 229 :
005B 230 LOADVEC ECC$REENABLE ;MEMORY ERROR TIMERS
```

0060	231	LOADVEC	EXESINIBOOTADP	;INIT BOOT DEVICE ADAPTER
0065	232	LOADVEC	EXESDUMPCPUREG	;DUMP CPU-SPECIFIC IPR'S INTO EMB
006A	233	LOADVEC	EXESREGRESTOR	;RESTORE CPU-SPECIFIC IPR'S
006F	234	LOADVEC	EXESREGSAVE	;SAVE CPU-SPECIFIC IPR'S
0074	235	LOADVEC	EXESINIPROCREG	;INIT PROCESSOR REGISTERS
0079	236	LOADVEC	EXESTEST_CSR	;TEST UB_CSR FOR EXISTENCE
007E	237	LOADVEC	IOCSPURGDATAP	;PURGE DATAPATH
0083	238	LOADVEC	INISMPMADP	;INITIALIZE MULTIPOINT MEMORY
0088	239	LOADVEC	EXESSTARTUPADP	;Startup up any adapters
008D	240	LOADVEC	EXESSHUTDOWNADP	;Shutdown any (all) adapters
0092	241	LOADVEC	MASRAVAIL	;Multiport memory
0097	242	LOADVEC	MASREQUEST	;Multiport memory request
009C	243	LOADVEC	MASINITIAL	;Multiport initialization
00A1	244	LOADVEC	CONSSTARTIO	;Console start I/O
00A6	245	LOADVEC	CONSSET_LINE	;Set line
00AB	246	LOADVEC	CONSDS_SET	;Data set
00B0	247	LOADVEC	CON\$XON	;XON to console
00B5	248	LOADVEC	CON\$XOFF	;XOFF to console
00BA	249	LOADVEC	CONSSTOP	;STOP output
00BF	250	LOADVEC	CONSSTOP2	;stop for 2 seconds
00C4	251	LOADVEC	CON\$ABORT	;Abort console I/O
00C9	252	LOADVEC	CON\$RESUME	;Resume output
00CE	253	LOADVEC	CONSSET MODEM	;Set modem
00D3	254	LOADVEC	CON\$NULL	;Null routine
00D8	255	LOADVEC	CON\$DISCONNECT	;Console disconnect routine
00DD	256	LOADVEC	CON\$INITIAL	;Initialize console controller
00E2	257	LOADVEC	CON\$INITLINE	;Initialize console line
00E7	258	LOADVEC	CON\$INTINP	;Input interrupt
00EC	259	LOADVEC	CON\$INTOUT	;Output interrupt
00F1	260	LOADVEC	CON\$SENDCONSCMD	;Send cpu-dependent command to console
00F6	261	LOADVEC	SYSL\$CLRSBIA, -	;Clear SBIA error bits
00FB	262		DEF RTN=EXESLOAD_NOP	
0100	263	LOADVEC	CON\$OWNCTY	;Set up to talk directly to console tty
0105	264	LOADVEC	CON\$RELEASECTY	;Restore normal console cty interface
010A	265	LOADVEC	CON\$GETCHAR	;Get a character from the console cty
010F	266	LOADVEC	CON\$PUTCHAR	;Put a character out to the console cty
0114	267	LOADVEC	CON\$INIT_CTY	;Initialization routine for the console cty
0119	268	LOADVEC	EXESREAD_TODR	;Read Time-Of-Day Register
011E	269	LOADVEC	EXESWRITE_TODR	;Write Time-Of-Day Register
0123	270	LOADVEC	EXESINIT_TODR	;Initialize system time-of-day
0128	271	LOADVEC	INISCONSLE	;Initialize console device data struc
012D	272	LOADVEC	EXESINI_TIMWAIT	;Initialize time-wait loop data cells
0132	273	LOADVEC	EXESREADP_TODR	;Read physical TODR register
0137	274	LOADVEC	EXESWRITEP_TODR	;Write physical TODR register
013C	275	LOADVEC	EXESMOUNTVER	;Mount verification main entry point
0141	276	LOADVEC	EXESMNTVERSIO	;Mount verification start I/O request
0146	277	LOADVEC	EXESMNTVERSHDOL	;Mount verification online shadow unit
014B	278	LOADVEC	EXESCLUTRANIO	;Mount verification VAXcluster state
0150	279			; transition block I/O
0150	280	LOADVEC	EXESUPDGNERNUM	;Mount verification update shadow set
0155	281			; generation number
015A	282	LOADVEC	EXESMNTVER_DVI_ASSIST	;Mount verification \$GETDVI escape
015F	283	LOADVEC	EXESMNTVERSP1	;Mount verification spare xfer vector
015F	284	LOADVEC	EXESMNTVERSP2	;Mount verification spare xfer vector
0164	285	LOADVEC	EXESGL_MVMSLBAS, 1,, -	;Mount verification MVMSL base address
	286		EXESAB_MVMSLBAS	
	287		:	

```
0164 288 : Extra packed vectors. Current target of these vectors in SYSLOA is a
0164 289 : halt instruction in ERRSUB*.
0164 290 :
0164 291 : LOADVEC EXE$EXTRA6 ;EXTRA LABEL
0169 292 : LOADVEC EXE$EXTRA7 ;EXTRA LABEL
016E 293 : LOADVEC EXE$EXTRA8 ;EXTRA LABEL
0173 294 : LOADVEC EXE$EXTRA9 ;EXTRA LABEL
0178 295 : LOADVEC EXE$EXTRA10 ;EXTRA LABEL
017D 296 :
017D 297 :
017D 298 : Define pointers to data structures.
017D 299 :
017D 300 :
017D 301 : LOADVEC EXE$MCHK_ERRCNT,TYPE=1 ;Point to array of mchk error counters.
0182 302 :
0182 303 : .IF DF,PRMSW ; For the loadable image
FFFFF0182 304 : .LONG -1 ; put in a stopper signal
0186 305 : .ENDC
0186 306 :
0186 307 :
0186 308 : IF LINKAGE WITH SYS.EXE, DEFINE A LOAD ERROR HANDLER AND A HANDLER THAT
0186 309 : RSB'S HARMLESSLY (FOR A ROUTINE USED BY XDELTA THAT MAY BE CALLED BEFORE
0186 310 : BEING LOADED).
0186 311 :
0186 312 :
0186 313 :
0186 314 : .IF NDF,PRMSW
0186 315 : EXE$LOAD_ERROR:: ;COME HERE IF CALL TO UNLOADED
0186 316 : HALT ; SUBROUTINE
0186 317 : EXE$LOAD_NOP:: ;COME HERE IF ROUTINE NOT LOADED
0186 318 : RSB ; BUT DON'T WANT TO ERROR HALT
0186 319 : .ENDC
0186 320 :
0186 321 :
0186 322 : IF LINKAGE WITH SYS.EXE, DEFINE THE DISPATCH VECTOR FOR LOADABLE CODE
0186 323 : SYSTEM SERVICE DISPATCHERS. THERE ARE SEPARATE VECTORS FOR EXEC AND
0186 324 : KERNEL MODE WITH TWO SPARE ENTRIES IN EACH.
0186 325 :
0186 326 :
0186 327 : .IF NDF,PRMSW
0186 328 :
0186 329 : EXE$LOAD_KDISP:: ;Kernel mode dispatchers
0186 330 : EXE$LOAD_KCJF::
0186 331 : JSB @#EXE$LOAD_NOP ;CJF
0186 332 : EXE$LOAD_KRUF::
0186 333 : JSB @#EXE$LOAD_NOP ;RUF
0186 334 : EXE$LOAD_KSPR1::
0186 335 : JSB @#EXE$LOAD_NOP ;First spare
0186 336 : EXE$LOAD_KSPR2::
0186 337 : JSB @#EXE$LOAD_NOP ;Second spare
0186 338 : RSB
0186 339 :
0186 340 : EXE$LOAD_EDISP:: ;Exec mode dispatchers
0186 341 : EXE$LOAD_ESPR1::
0186 342 : JSB @#EXE$LOAD_NOP ;First spare
0186 343 : EXE$LOAD_ESPR2::
0186 344 : JSB @#EXE$LOAD_NOP ;Second spare
```

```

0186 345 RSB
0186 346
0186 347 .ENDC
0186 348
0186 349 ;
0186 350 ; IF LINKAGE WITH SYSLOAxxx.EXE, DEFINE 15 LONGWORDS OF PATCH AREA:
0186 351 ;
0186 352
0186 353 .IF DF,PRMSW
00000000 354 .PSECT _PATCH
0000 355 PATCH_DESC::
0000003C 0000 356 .LONG 15*4
00000008' 0004 357 .LONG PATCH_AREA
0008 358 PATCH_AREA::
00000044 0008 359 .BLKL 15
0044 360 .ENDC
0044 361
0044 362 .END

```

CONSABORT
 CONSDISCONNECT
 CONSDS_SET
 CON\$GETCHAR
 CON\$INITIAL
 CON\$INITLINE
 CON\$INIT CTY
 CON\$INTINP
 CON\$INTOUT
 CON\$NULL
 CON\$OWNCTY
 CON\$PUTCHAR
 CON\$RELEASECTY
 CON\$RESUME
 CON\$SENDCONSCMD
 CON\$SET_LINE
 CON\$SET_MODEM
 CON\$STARTIO
 CON\$STOP
 CON\$STOP2
 CON\$XOFF
 CON\$XON
 DYN\$C_LOADCODE
 ECC\$REENABLE
 EXE\$AB_MVMSLBAS
 EXE\$CLUTRANIO
 EXE\$DUMPCPUREG
 EXE\$EXTRA1
 EXE\$EXTRA10
 EXE\$EXTRA2
 EXE\$EXTRA3
 EXE\$EXTRA4
 EXE\$EXTRA5
 EXE\$EXTRA6
 EXE\$EXTRA7
 EXE\$EXTRA8
 EXE\$EXTRA9
 EXE\$INIBOOTADP
 EXE\$INIPROCREG
 EXE\$INIT TODR
 EXE\$INI_TIMWAIT
 EXE\$INT54
 EXE\$INT58
 EXE\$INT5C
 EXE\$INT60
 EXE\$MCHK
 EXE\$MCHK_ERRCNT
 EXE\$MNTVERSHDOL
 EXE\$MNTVERSIO
 EXE\$MNTVERS P1
 EXE\$MNTVERS P2
 EXE\$MNTVER DVI_ASSIST
 EXE\$MOUNTVER
 EXE\$READP TODR
 EXE\$READ TODR
 EXE\$REGRESTOR
 EXE\$REGSAVE

[illegible]

```

EXESSHUTDOWNADP
EXESSTARTUPADP
EXESTEST CSR
EXESUPDGNERNUM
EXESWRITEP TODR
EXESWRITE TODR
INISCONSOLE
INISIOMAP
INISMPHADP
IOCSPURGDATAP
MASINITIAL
MASRAVAIL
MASREQUEST
PATCH_AREA
PATCH_DESC
PRMSW-
PRTSC_ER
PRTSC_EW
SYSL$BEGIN
SYSL$CLRSBIA
SYSL$END
UBA_UNEXINT

```

*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
*****	X	03
00000008	RG	04
00000000	RG	04
== 00000001		
00000007		
00000005		
00000000	RG	03
*****	X	03
00000000	RG	02
*****	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
LOAD_END	00000000 (0.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC QUAD
\$\$\$000	00000186 (390.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
_PATCH	00000044 (68.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.26
Command processing	107	00:00:00.72	00:00:01.42
Pass 1	201	00:00:06.39	00:00:07.75
Symbol table sort	0	00:00:00.28	00:00:00.28
Pass 2	83	00:00:01.76	00:00:02.10
Symbol table output	10	00:00:00.06	00:00:00.06
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	435	00:00:09.34	00:00:11.92

The working set limit was 1200 pages.

31911 bytes (63 pages) of virtual memory were used to buffer the intermediate code.

There were 20 pages of symbol table space allocated to hold 238 non-local and 2 local symbols.

363 source lines were read in Pass 1, producing 22 object records in Pass 2.

11 pages of virtual memory were used to define 9 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	3
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	7

324 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:LOAVEC/OBJ=OBJ\$:LOAVEC MSRC\$:PRMSW/UPDATE=(ENH\$:PRMSW)+MSRC\$:SYSLOAVEC/UPDATE=(ENH\$:SYSLOAVEC)+EXECMLS/LIR

0377

**DIGITAL
CONFIDE**

EQUIPMENT
INITIAL AND

CORPORATION
PROPRIETARY