```
LLL                OOOOOOOOO          AAAAAAAAA      DDDDDDDDDDD          SSSSSSSSSSS      SSSSSSSSSSS
LLL                OOOOOOOOO          AAAAAAAAA      DDDDDDDDDDD          SSSSSSSSSSS      SSSSSSSSSSS
LLL                OOOOOOOOO          AAAAAAAAA      DDDDDDDDDDD          SSSSSSSSSSS      SSSSSSSSSSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD    SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD    SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD    SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD    SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD    SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD    SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD       SSSSSSSSS        SSSSSSSSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD       SSSSSSSSS        SSSSSSSSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD       SSSSSSSSS        SSSSSSSSS
LLL              OOO         OOO    AAAAAAAAAAAAAAA   DDD         DDD                SSS              SSS
LLL              OOO         OOO    AAAAAAAAAAAAAAA   DDD         DDD                SSS              SSS
LLL              OOO         OOO    AAAAAAAAAAAAAAA   DDD         DDD                SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD                SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD                SSS              SSS
LLL              OOO         OOO    AAA         AAA   DDD         DDD                SSS              SSS
LLLLLLLLLLLLLLL     OOOOOOOOO       AAA         AAA   DDDDDDDDDDD        SSSSSSSSSSS      SSSSSSSSSSS
LLLLLLLLLLLLLLL     OOOOOOOOO       AAA         AAA   DDDDDDDDDDD        SSSSSSSSSSS      SSSSSSSSSSS
LLLLLLLLLLLLLLL     OOOOOOOOO       AAA         AAA   DDDDDDDDDDD        SSSSSSSSSSS      SSSSSSSSSSS
```

```
RRRRRRRR    DDDDDDDD    BBBBBBBB    DDDDDDDD      IIIIII      SSSSSSSS  PPPPPPPP
RRRRRRRR    DDDDDDDD    BBBBBBBB    DDDDDDDD      IIIIII      SSSSSSSS  PPPPPPPP
RR      RR  DD      DD  BB      BB  DD      DD      II      SS            PP      PP
RR      RR  DD      DD  BB      BB  DD      DD      II      SS            PP      PP
RR      RR  DD      DD  BB      BB  DD      DD      II      SS            PP      PP
RR      RR  DD      DD  BB      BB  DD      DD      II      SS            PP      PP
RRRRRRRR    DD      DD  BBBBBBBB    DD      DD      II        SSSSSS      PPPPPPPP
RRRRRRRR    DD      DD  BBBBBBBB    DD      DD      II        SSSSSS      PPPPPPPP
RR   RR     DD      DD  BB      BB  DD      DD      II              SS    PP
RR   RR     DD      DD  BB      BB  DD      DD      II              SS    PP
RR    RR    DD      DD  BB      BB  DD      DD      II              SS    PP                ....
RR    RR    DD      DD  BB      BB  DD      DD      II              SS    PP                ....
RR     RR   DDDDDDDD    BBBBBBBB    DDDDDDDD      IIIIII    SSSSSSSS    PP                ....
RR     RR   DDDDDDDD    BBBBBBBB    DDDDDDDD      IIIIII    SSSSSSSS    PP                ....

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          .TITLE  RDBDISP - Rights database system service dispatcher
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 ; Facility: Rights database (RDB) System Services
0000    29 ;++
0000    30 ; Abstract:
0000    31 ;       This module contains the dispatcher for the rights database
0000    32 ;       loadable system services. These system services are implemented
0000    33 ;       in a privileged shareable image. The remaining rights database
0000    34 ;       system services are in the exec.
0000    35 ;
0000    36 ; Overview:
0000    37 ;       The rights database system services are contained in a privileged
0000    38 ;       shareable image that is linked into user program images in exactly the
0000    39 ;       same fashion as any shareable image. The creation and installation of
0000    40 ;       a privileged, shareable image is slightly different from that of an
0000    41 ;       ordinary shareable image.  These differences are:
0000    42 ;
0000    43 ;               1. A vector defining the entry points and providing other
0000    44 ;                  control information to the image activator.  This vector
0000    45 ;                  is a the lowest address in an image section with the VEC
0000    46 ;                  attribute.
0000    47 ;
0000    48 ;               2. The shareable image is linked with the /PROTECT option
0000    49 ;                  that marks all of the image sections so that they will
0000    50 ;                  protected and given EXEC mode ownership by the image
0000    51 ;                  activator.
0000    52 ;
0000    53 ;               3. The shareable image MUST be installed /SHARE /PROTECT
0000    54 ;                  with the INSTALL utility in order for the image activator
0000    55 ;                  to connect the privileged shareable image to the change mode
0000    56 ;                  dispatchers.
0000    57 ;
```

```
0000     58 ;       A privileged shareable image implementing user written system services
0000     59 ;       is comprised of the following major components:
0000     60 ;
0000     61 ;               1. A transfer vector containing all of the entry points and
0000     62 ;                  collecting them at the lowest virtual address in the shareable
0000     63 ;                  image.  This formalism enables revision of the shareable
0000     64 ;                  image without necessitating the relinking of images that
0000     65 ;                  use it.
0000     66 ;
0000     67 ;               2. A Privileged Library Vector in a PSECT with the VEC attribute
0000     68 ;                  that describes the entry points for dispatching EXEC and
0000     69 ;                  KERNEL mode services along with validation information.
0000     70 ;
0000     71 ;               3. A dispatcher for kernel mode services.  This code will
0000     72 ;                  be called by the VMS change mode dispatcher when it
0000     73 ;                  fails to recognize a kernel mode service request.
0000     74 ;
0000     75 ;               4. A dispatcher for executive mode services.  This code will
0000     76 ;                  be called by the VMS change mode dispatcher when it fails
0000     77 ;                  to recognize an executive mode service request.
0000     78 ;
0000     79 ;               5. Service routines to perform the various services.
0000     80 ;
0000     81 ; Enviornment:
0000     82 ;
0000     83 ;       VAX/VMS operating system, installed as a privileged shareable image
0000     84 ;
0000     85 ;--
0000     86 ;
0000     87 ; Author:
0000     88 ;
0000     89 ;       R. Scott Hanna, CREATION DATE: 17-JAN-1983
0000     90 ;
0000     91 ; Modified by:
0000     92 ;
0000     93 ;       V03-005 ACG0417         Andrew C. Goldstein,    18-Apr-1984  16:35
0000     94 ;               Add rundown routine to dequeue ACL locks; fix minimum
0000     95 ;               arg count for $FORMAT_ACL and $PARSE_ACL.
0000     96 ;
0000     97 ;       V03-004 LY0468          Larry Yetto             22-MAR-1984 14:11
0000     98 ;               Add two new paramaters to $MOD_IDENT for the new name
0000     99 ;               and new value.
0000    100 ;
0000    101 ;       V03-003 LMP0185         L. Mark Pilant,         6-Feb-1984  12:55
0000    102 ;               Add a new service $CHANGE_ACL for changing the ACL on
0000    103 ;               random objects.
0000    104 ;
0000    105 ;       V03-002 LMP0100         L. Mark Pilant,         14-Apr-1983  11:32
0000    106 ;               Add new services: $FORMAT_ACL and $PARSE_ACL.
0000    107 ;
0000    108 ;       V03-001 RSH0009         R. Scott Hanna          09-Mar-1983
0000    109 ;               Changed CHME codes.
0000    110 ;
0000    111 ;--
```

```
                        .SBTTL  Declarations and Equates
                ;
                ;       Macro Definitions
                ;
                ;       DEFINE_SERVICE - A macro to make the appropriate entries in several
                ;                        different PSECTs required to define an EXEC or KERNEL
                ;                        mode service.  These include the transfer vector,
                ;                        the case table for dispatching, and a table containing
                ;                        the number of required arguments.
                ;
                ;       DEFINE_SERVICE Name,Number_of_Arguments,Mode
                ;
                        .MACRO  DEFINE_SERVICE,NAME,NARG=0,MODE=KERNEL
                        .PSECT  $$$TRANSFER_VECTOR,PAGE,NOWRT,EXE,PIC,SHR
                        .ALIGN  QUAD                    ; Align entry points for speed and style
                        .TRANSFER       NAME            ; Define name as universal symbol for entry
                        .MASK   NAME                    ; Use entry mask defined in main routine
                        .IF     IDN MODE,KERNEL
                        CHMK    #<KCODE_BASE+KERNEL_COUNTER> ; Change to kernel mode and execute
                        RET                             ; Return
                        KERNEL_COUNTER=KERNEL_COUNTER+1 ; Advance counter
                
                        .PSECT  KERNEL_NARG,BYTE,NOWRT,EXE,PIC,SHR
                        .BYTE   NARG                    ; Define number of required arguments
                
                        .PSECT  USER_KERNEL_DISP1,BYTE,NOWRT,EXE,PIC,SHR
                        .WORD   2+NAME-KCASE_BASE       ; Make entry in kernel mode CASE table
                
                        .IFF
                        CHME    #<ECODE_BASE+EXEC_COUNTER> ; Change to executive mode and execute
                        RET                             ; Return
                        EXEC_COUNTER=EXEC_COUNTER+1     ; Advance counter
                
                        .PSECT  EXEC_NARG,BYTE,NOWRT,EXE,PIC,SHR
                        .BYTE   NARG                    ; Define number of required arguments
                
                        .PSECT  USER_EXEC_DISP1,BYTE,NOWRT,EXE,PIC,SHR
                        .WORD   2+NAME-ECASE_BASE       ; Make entry in exec mode CASE table
                        .ENDC                           ;
                        .ENDM   DEFINE_SERVICE          ;
                ;
                ;       Equated Symbols
                ;
                
                        $PLVDEF                         ; Define PLV offsets and values
                        $PRDEF                          ; Define processor register numbers
                ;
                ;       Initialize counters for change mode dispatching codes
                ;
00000000        KERNEL_COUNTER=0                        ; Kernel code counter
00000000        EXEC_COUNTER=0                          ; Exec code counter
                
                ;
                ;       Own Storage
                ;
                        .PSECT  KERNEL_NARG,BYTE,NOWRT,EXE,PIC,SHR
KERNEL_NARG:                                            ; Base of byte table containing the
```

```
                    170                                     ;  number of required arguments.
00000000            171             .PSECT  EXEC_NARG,BYTE,NOWRT,EXE,PIC,SHR
                    172 EXEC_NARG:                          ; Base of byte table containing the
                    173                                     ;  number of required arguments.
```

```
                        .SBTTL  Transfer Vector and Service Definitions
                ;++
                ; The use of transfer vectors to effect entry to the user written system services
                ; enables some updating of the shareable image containing them without necessitating
                ; a re-link of all programs that call them.  The PSECT containinng the transfer
                ; vector will be positioned at the lowest virtual address in the shareable image
                ; and so long as the transfer vector is not re-ordered, programs linked with
                ; one version of the shareable image will continue to work with the next.
                ;
                ; Thus as additional services are added to a privileged shareable image, their
                ; definitions should be added to the end of the following list to ensure that
                ; programs using previous versions of it will not need to be re-linked.
                ; To completely avoid relinking existing programs the size of the privileged
                ; shareable image must not change so some padding will be required to provide
                ; the opportunity for future growth.
                ;--

                        DEFINE_SERVICE  SYS$ADD_HOLDER,3,EXEC   ;Add Holder Record To The Rights Dat
                        DEFINE_SERVICE  SYS$ADD_IDENT,4,EXEC    ;Add Identifier To The Rights Databa
                        DEFINE_SERVICE  SYS$CREATE_RDB,1,EXEC   ;Create The Rights Database
                        DEFINE_SERVICE  SYS$FIND_HELD,4,EXEC    ;Find Identifiers Held By Holder
                        DEFINE_SERVICE  SYS$FIND_HOLDER,4,EXEC  ;Find Holder Of Identifier
                        DEFINE_SERVICE  SYS$MOD_HOLDER,4,EXEC   ;Modify Holder Record In Rights Data
                        DEFINE_SERVICE  SYS$MOD_IDENT,5,EXEC    ;Modify Identifier Record In Rights
                        DEFINE_SERVICE  SYS$REM_HOLDER,2,EXEC   ;Remove Holder Record From Rights Da
                        DEFINE_SERVICE  SYS$REM_IDENT,1,EXEC    ;Remove Identifier From Rights Datab

                ; ACL manipulation services.

                        DEFINE_SERVICE  SYS$FORMAT_ACL,7,EXEC   ;Convert ACE to text
                        DEFINE_SERVICE  SYS$PARSE_ACL,4,EXEC    ;Convert ACE to binary
                        DEFINE_SERVICE  SYS$CHANGE_ACL,7,EXEC   ;modify an object's ACL

                ;
                ; The base values used to generate the dispatching codes should be negative for
                ; user services and must be chosen to avoid overlap with any other privileged
                ; shareable images that will be used concurrently.  Their definition is
                ; deferred to this point in the assembly to cause their use in the preceding
                ; macro calls to be forward references that guarantee the size of the change
                ; mode instructions to be four bytes.  This satisfies an assumption that is
                ; made by for services that have to wait and be retried.  The PC for retrying
                ; the change mode instruction that invokes the service is assumed to be 4 bytes
                ; less than that saved in the change mode exception frame.  Of course, the particula
                ; service routine determines whether this is possible.
                ;
                ; The rights database system services have a block of 16 CHME codes reserved
                ; (16416-16431).
                ;

00000000                KCODE_BASE=0                            ; No CHMK codes used
00004020                ECODE_BASE=16416                        ; Base CHME code value for these services
```

```
                        .SBTTL  Change Mode Dispatcher Vector Block
                ;++
                ; This vector is used by the image activator to connect the privileged shareable
                ; image to the VMS change mode dispatcher.  The offsets in the vector are self-
                ; relative to enable the construction of position independent images.  The system
                ; version number will be used by the image activator to verify that this shareable
                ; image was linked with the symbol table for the current system.
                ;
                ;                               Change Mode Vector Format
                ;
                ;       +-----------------------------------------+
                ;       |           Vector Type Code              |   PLV$L_TYPE
                ;       |          (PLV$C_TYP_CMOD)               |
                ;       +-----------------------------------------+
                ;       |         System Version Number           |   PLV$L_VERSION
                ;       |           (SYS$K_VERSION)               |
                ;       +-----------------------------------------+
                ;       |     Kernel Mode Dispatcher Offset       |   PLV$L_KERNEL
                ;       |                                         |
                ;       +-----------------------------------------+
                ;       |        Exec Mode Entry Offset           |   PLV$L_EXEC
                ;       |                                         |
                ;       +-----------------------------------------+
                ;       |      User Rundown Service Offset        |   PLV$L_USRUNDWN
                ;       |                                         |
                ;       +-----------------------------------------+
                ;       |                Reserved                 |
                ;       |                                         |
                ;       +-----------------------------------------+
                ;       |          RMS Dispatcher Offset          |   PLV$L_RMS
                ;       |                                         |
                ;       +-----------------------------------------+
                ;       |             Address Check               |   PLV$L_CHECK
                ;       |                                         |
                ;       +-----------------------------------------+
                ;
00000000        .PSECT  USER_SERVICES,PAGE,VEC,PIC,NOWRT,EXE,SHR
                .LONG   PLV$C_TYP_CMOD          ; Set type of vector to change mode dispatch
00000000'       .LONG   SYS$K_VERSION           ; Identify system version
00000000        .LONG   0                       ; No kernel mode dispatcher
00000001'       .LONG   EXEC_DISPATCH-.         ; Offset to executive mode dispatcher
FFFFFFF4'       .LONG   RDB_RUNDOWN-.           ; Rundown routine
00000000        .LONG   0                       ; Reserved.
00000000        .LONG   0                       ; No RMS dispatcher
00000000        .LONG   0                       ; Address check - PIC image
```

```
                     0020   275          .SBTTL  Executive Mode Dispatcher
                     0020   276 ;++
                     0020   277 ; Input Parameters:
                     0020   278 ;
                     0020   279 ;       (SP) - Return address if bad change mode value
                     0020   280 ;
                     0020   281 ;        R0  - Change mode argument value.
                     0020   282 ;
                     0020   283 ;        AP  - Argument pointer existing when the change
                     0020   284 ;              mode instruction was executed.
                     0020   285 ;
                     0020   286 ;        FP  - Address of minimal call frame to exit
                     0020   287 ;              the change mode dispatcher and return to
                     0020   288 ;              the original mode.
                     0020   289 ;--
                00000000    290          .PSECT  USER_EXEC_DISP0,BYTE,NOWRT,EXE,PIC,SHR
                     0000   291 EACCVIO:                                ; Exec access violation
     50  0000'8F    3C  0000   292          MOVZWL  #SS$_ACCVIO,R0      ; Set access violation status code
                     04  0005   293          RET                         ;  and return
                     0006   294 EINSFARG:                               ; Exec insufficient arguments.
     50  0000'8F    3C  0006   295          MOVZWL  #SS$_INSFARG,R0     ; Set status code and
                     04  000B   296          RET                         ;  return
                     05  000C   297 ENOTME: RSB                         ; RSB to forward request
                     000D   298
                     000D   299 EXEC_DISPATCH::                         ; Entry to dispatcher
     51  BFE0 CO    9E  000D   300          MOVAB   W^-ECODE_BASE(R0),R1 ; Normalize dispatch code value
            F8      19  0012   301          BLSS    ENOTME              ; Branch if code value too low
         0C 51      B1  0014   302          CMPW    R1,#EXEC_COUNTER    ; Check high limit
            F3      1E  0017   303          BGEQU   ENOTME              ; Branch if out of range
                     0019   304 ;
                     0019   305 ; The dispatch code has now been verified as being handled by this dispatcher,
                     0019   306 ; now the argument list will be probed and the required number of arguments
                     0019   307 ; verified.
                     0019   308 ;
         51   0000'CF41    9A  0019   309          MOVZBL  W^EXEC_NARG[R1],R1  ; Get required argument count
     51   00000004 9F41    DE  001F   310          MOVAL   @#4[R1],R1          ; Compute byte count including arg count
                     0027   311          IFNORD  R1,(AP),EACCVIO     ; Branch if arglist not readable
        BFE0'CF40    6C  91  002D   312          CMPB    (AP),W^<EXEC_NARG-ECODE_BASE>[R0] ; Check for required number
                     D1  1F  0033   313          BLSSU   EINSFARG            ;  of arguments
                5E   5D  D0  0035   314          MOVL    FP,SP               ; remove JSB longwords from stack
                     50  AF  0038   315          CASEW   R0,-                ; Case on change mode
                     003A   316                  -                           ; argument value
                     003A   317                  #ECODE_BASE,-               ; Base value
          0B   4020 8F       003A   318                  #<EXEC_COUNTER-1>   ; Limit value (number of entries)
                     003E   319 ECASE_BASE:                             ; Case table base address for DEFINE_SERVICE
                     003E   320 ;
                     003E   321 ;       Case table entries are made in the PSECT USER_EXEC_DISP1 by
                     003E   322 ;       invocations of the DEFINE_SERVICE macro.  The three PSECTS,
                     003E   323 ;       USER_EXEC_DISP0,1,2 will be abutted in lexical order at link-time.
                     003E   324 ;
                00000000    325          .PSECT  USER_EXEC_DISP2,BYTE,NOWRT,EXE,PIC,SHR
                     0000   326          BUG_CHECK IVSSRVRQST,FATAL          ; range validation performed above so
                     0004   327                                              ; should never reach here
```

```
                     0004    329          .SBTTL  Rundown Handler
                     0004    330
                     0004    331 ;++
                     0004    332 ;
                     0004    333 ; The rundown handler is called when image or process rundown is performed.
                     0004    334 ; It cleans up context related to this package of services.
                     0004    335 ;
                     0004    336 ; Calling Sequence:
                     0004    337 ;
                     0004    338 ;       JSB RDB_RUNDOWN
                     0004    339 ;
                     0004    340 ; Inputs:
                     0004    341 ;
                     0004    342 ;       none
                     0004    343 ;
                     0004    344 ;--
                     0004    345
                     0004    346
                     0004    347 RDB_RUNDOWN:
00000000'EF    00    FB 0004    348          CALLS   #0,RUNDOWN_CHANGE_ACL   ; Do rundown for the $CHANGE_ACL service
                     05 000B    349          RSB
                     000C    350
                     000C    351
                     000C    352          .END
```

```
BUG$_IVSSRVRQST                 ********    X   08
EACCVIO                         00000000 R      07
ECASE_BASE                      0000003E R      07
ECODE_BASE                    = 00004020
EINSFARG                        00000006 R      07
ENOTME                          0000000C R      07
EXEC_COUNTER                  = 0000000C
EXEC_DISPATCH                   0000000D RG     07
EXEC_NARG                       00000000 R      03
KCODE_BASE                    = 00000000
KERNEL_COUNTER                = 00000000
KERNEL_NARG                     00000000 R      02
PLV$C_TYP_CMOD                = 00000001
RDB_RUNDOWN                     00000004 R      08
RUNDOWN_CHANGE_ACL              ********    X   08
SS$_ACCVIO                      ********    X   07
SS$_INSFARG                     ********    X   07
SYS$ADD_HOLDER                  ********    X   04
SYS$ADD_IDENT                   ********    X   04
SYS$CHANGE_ACL                  ********    X   04
SYS$CREATE_RDB                  ********    X   04
SYS$FIND_HELD                   ********    X   04
SYS$FIND_HOLDER                 ********    X   04
SYS$FORMAT_ACL                  ********    X   04
SYS$K_VERSION                   ********    X   06
SYS$MOD_HOLDER                  ********    X   04
SYS$MOD_IDENT                   ********    X   04
SYS$PARSE_ACL                   ********    X   04
SYS$REM_HOLDER                  ********    X   04
SYS$REM_IDENT                   ********    X   04
```

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
KERNEL_NARG	00000000	(0.)	02	(2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC BYTE
EXEC_NARG	0000000C	(12.)	03	(3.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC BYTE
$$$TRANSFER_VECTOR	0000005F	(95.)	04	(4.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC PAGE
USER_EXEC_DISP1	00000018	(24.)	05	(5.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC BYTE
USER_SERVICES	00000020	(32.)	06	(6.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	VEC PAGE
USER_EXEC_DISP0	0000003E	(62.)	07	(7.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC BYTE
USER_EXEC_DISP2	0000000C	(12.)	08	(8.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.07	00:00:00.76
Command processing	133	00:00:00.68	00:00:04.62
Pass 1	150	00:00:02.62	00:00:08.05
Symbol table sort	0	00:00:00.12	00:00:00.12

Pass 2	83	00:00:00.95	00:00:02.41
Symbol table output	5	00:00:00.03	00:00:00.24
Psect synopsis output	3	00:00:00.05	00:00:00.10
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	408	00:00:04.54	00:00:16.39

The working set limit was 1200 pages.
11621 bytes (23 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 120 non-local and 0 local symbols.
352 source lines were read in Pass 1, producing 27 object records in Pass 2.
14 pages of virtual memory were used to define 11 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	2
_255DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	7

163 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

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

LNKUMCTRL
LIS

RDBSHR
LIS

SYSACLSRV
LIS

RDBDISP
LIS

LOADSS

SECURESHR
MAP

FINDHELD
LIS