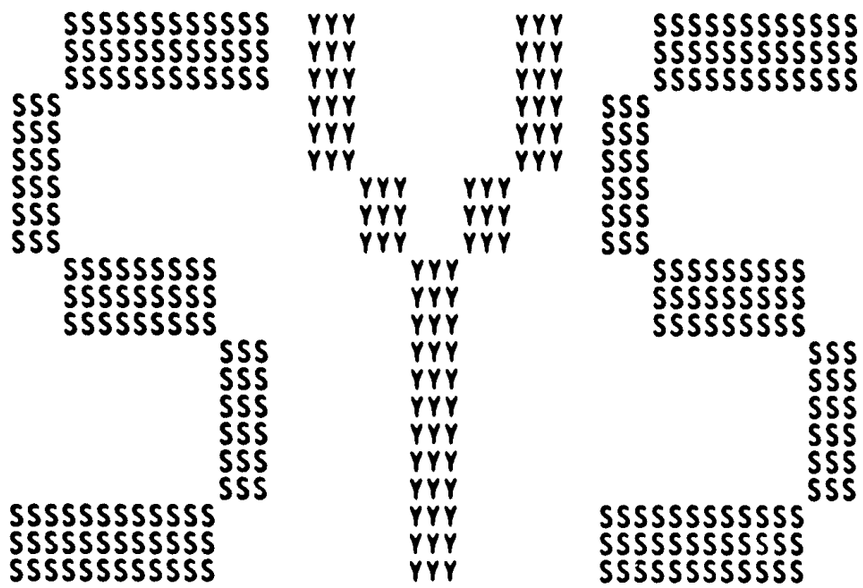

```
  SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  YY      YY    IIIIII
  SSSSSSSS  YY      YY    SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS  YY      YY    IIIIII
SS          YY      YY  SS          GG          EE              TT      SS          YY      YY      II
SS          YY      YY  SS          GG          EE              TT      SS          YY      YY      II
SS            YY  YY    SS          GG          EE              TT      SS            YY  YY        II
SS            YY  YY    SS          GG          EE              TT      SS            YY  YY        II
  SSSSSS        YY        SSSSSS    GG          EEEEEEEE        TT        SSSSSS        YY          II
  SSSSSS        YY        SSSSSS    GG          EEEEEEEE        TT        SSSSSS        YY          II
        SS      YY              SS  GG  GGGGGG  EE              TT              SS      YY          II
        SS      YY              SS  GG  GGGGGG  EE              TT              SS      YY          II
        SS      YY              SS  GG      GG  EE              TT              SS      YY          II          ....
        SS      YY              SS  GG      GG  EE              TT              SS      YY          II          ....
SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE      TT      SSSSSSSS        YY        IIIIII        ....
SSSSSSSS        YY      SSSSSSSS      GGGGGG    EEEEEEEEEE      TT      SSSSSSSS        YY        IIIIII        ....

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  SYSGETSYI - GET SYSTEM INFORMATION SYSTEM SERVICE
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: VMS Executive, System services.
0000    29  ;
0000    30  ; ABSTRACT:
0000    31  ;
0000    32  ;       Return processor information to caller, specifically
0000    33  ;       processor ID register, processor type, and VMS version number.
0000    34  ;
0000    35  ; ENVIRONMENT: Kernel Mode
0000    36  ;
0000    37  ; AUTHOR: John A. Ywoskus,       CREATION DATE: 06-August-1981
0000    38  ;
0000    39  ; MODIFIED BY:
0000    40  ;
0000    41  ;       V03-026 CWH3026         CW Hobbs                23-Jul-1984
0000    42  ;               Treat the QUANTUM item as special, since it is
0000    43  ;               stored as a negative number.
0000    44  ;
0000    45  ;       V03-025 MSH0059         Michael S. Harvey       3-Jul-1984
0000    46  ;               Treat a specified CSID argument of zero as if the argument
0000    47  ;               hadn't been specified at all. In either case, there is no
0000    48  ;               CSID value specified and the behavior of GETSYI should be
0000    49  ;               the same for both.
0000    50  ;
0000    51  ;       V03-024 MSH0021         Michael S. Harvey       9-Mar-1984
0000    52  ;               Allow access to SYI items stashed away in the local SB,
0000    53  ;               regardless of whether we're in a cluster or not.
0000    54  ;
0000    55  ;       V03-023 MSH0013         Michael S. Harvey       2-Mar-1984
0000    56  ;               Correctly extract node name length so as not to clobber
0000    57  ;               P1 space and crash the system.
```

```
0000     58 ;
0000     59 ;       V03-022 WMC0003         Wayne Cardoza           9-Feb-1984
0000     60 ;               Add $ARCDEF.
0000     61 ;
0000     62 ;       V03-021 WMC0002         Wayne Cardoza           29-Jan-1984
0000     63 ;               Add F and G floating flags.
0000     64 ;
0000     65 ;       V03-020 KPL0001         Peter Lieberwirth       15-Jan-1984
0000     66 ;               Fix typo in V03-019.
0000     67 ;
0000     68 ;       V03-019 WMC0001         Wayne Cardoza           07-Jan-1984
0000     69 ;               Add page and swap file sizes.
0000     70 ;
0000     71 ;       V03-018 TCM0001         Trudy C. Matthews       28-Dec-1983
0000     72 ;               In EXE$NAMCSID, do not access node name passed by caller
0000     73 ;               after raising IPL (it may be pagable).  Make QUORUM a
0000     74 ;               special parameter; it is stored as a negative value but should
0000     75 ;               be displayed as a positive one.
0000     76 ;
0000     77 ;       V03-017 KFH0011         Ken Henderson           30 Aug 1983
0000     78 ;               Fix resetting of IPL on error path.
0000     79 ;               Add documentation of how
0000     80 ;               itemcodes are added.
0000     81 ;
0000     82 ;       V03-016 KFH0010         Ken Henderson           23 Aug 1983
0000     83 ;               Fix checking of item code validity.
0000     84 ;               Update max structure code.
0000     85 ;
0000     86 ;       V03-015 KFH0009         Ken Henderson           18 Aug 1983
0000     87 ;               Made SCS_EXISTS special and boolean.
0000     88 ;
0000     89 ;       V03-014 KFH0008         Ken Henderson           28 Jul 1983
0000     90 ;               Finished support for 'retired' item-codes.
0000     91 ;               Took out call to SCS$CONFIG_SYS.
0000     92 ;
0000     93 ;       V03-013 KFH0007         Ken Henderson           12 Jul 1983
0000     94 ;               Added temporary additional check for
0000     95 ;               clusterness.
0000     96 ;
0000     97 ;       V03-012 KFH0006         Ken Henderson           26 May 1983
0000     98 ;               Changed EXE$NAMCSID entry point to
0000     99 ;               be non-Global.
0000    100 ;
0000    101 ;       V03-011 KFH0005         Ken Henderson           25 May 1983
0000    102 ;               Updated code to use IFCLSTR and IFNOCLSTR.
0000    103 ;
0000    104 ;       V03-010 KFH0004         Ken Henderson           21 May 1983
0000    105 ;               Added support for wild-carding through
0000    106 ;               nodes. Added NAMCSID and EXE$NAMCSID routines.
0000    107 ;               Cleaned up usage of LOCAL_SPACE on stack.
0000    108 ;
0000    109 ;       V03-009 KFH0003         Ken Henderson           11 Mar 1983
0000    110 ;               Added .WARN if item-code is undefined.
0000    111 ;
0000    112 ;       V03-008 KFH0002         Ken Henderson           25 Feb 1983
0000    113 ;               Added definition of GETSYISW.
0000    114 ;
```

```
0000  ;       V03-007 KFH0001         Ken Henderson           16 Feb 1983
0000  ;               Major rewrite of EXE$GETSYI and related
0000  ;               routines to make it table-driven like GETJPI,
0000  ;               and allow for SYSBOOT parameters and other
0000  ;               enhancements.
0000  ;
0000  ;       V03-006 MSH0001         Maryann Hinden          23-Mar-1982
0000  ;               Fix broken BSBW.
0000  ;
0000  ;       V03-005 JAY0006         John A. Ywoskus         17-Mar-1982
0000  ;               Change SS$_EXQUOTA return error to SS$_EXASTLM.
0000  ;
0000  ;       V03-004 JAY0005         John A. Ywoskus         21-Jan-1982
0000  ;               Return 8 bytes for system version instead of 4.
0000  ;               General cleanup.
0000  ;
0000  ;       V02-003 LJK0082         Lawrence J. Kenah       11-Nov-1981
0000  ;               Write accessibility of multiple page buffer can
0000  ;               now be done on global routine.
0000  ;
0000  ;       V03-002 JAY0004         John A. Ywoskus         05-Oct-1981
0000  ;               Add null arguments so call list is compatable with
0000  ;               $GETJPI. Also, make external references be
0000  ;               addressed with G^, and include VA and PSL defs.
0000  ;
0000  ;       V03-001 JAY0003         John A. Ywoskus         08-Sep-1981
0000  ;               Fix null item bug, make return length optional.
0000  ;--
0000  ;
```

```
0000   145 ;                GUIDE TO GETJPI/GETSYI/GETDVI
0000   146 ;                ----------------------------
0000   147 ;
0000   148 ;Overview
0000   149 ;--------
0000   150 ;
0000   151 ;These three system services are table-driven. The macro definition files
0000   152 ;that help define their tables are shared with DCL and the RTL. This results
0000   153 ;in new item-codes becoming useable with DCL's F$GETXXI lexical functions and
0000   154 ;the RTL's LIB$GETXXI routines automatically. Additionally, new SYSBOOT
0000   155 ;parameters become item-codes to the GETSYIs.
0000   156 ;
0000   157 ;The macro definition files are called JPITABLE.MAR, SYITABLE.MAR, and
0000   158 ;DVITABLE.MAR, and live in MASD$:<VMSLIB.SRC>. During a systembuild, they
0000   159 ;are inserted into the library SYS$LIBRARY:SYSBLDMLB.MLB. DCL and the RTL
0000   160 ;and SYS use this library to define their GETXXI tables. The system
0000   161 ;parameter file <SYS.SRC>SYSPARAM.MAR has also been conditionalized to be
0000   162 ;used to define GETSYI item-codes and is also inserted into SYSBLDMLB.MLB.
0000   163 ;
0000   164 ;
0000   165 ;NOTE:  SYSBLDMLB.MLB is a general macro library for holding macro
0000   166 ;       definitions that are shared between facilities, but will not
0000   167 ;       ship to the customer.
0000   168 ;
0000   169 ;
0000   170 ;When adding an item-code, at least two files need to be edited. One of the
0000   171 ;macro files listed above, as well as an SDL file that defines the 16-bit
0000   172 ;number which is the user-visible item-code. Also, if a SYSBOOT parameter is
0000   173 ;added, an SDL file needs to be updated to define the new GETSYI item-code.
0000   174 ;
0000   175 ;The GETDVI service actually uses only one table, but the GETSYI and GETJPI
0000   176 ;services use several. The JPITABLE file defines all the tables for GETJPI
0000   177 ;and the SYITABLE file defines all the tables for GETSYI. The different
0000   178 ;tables group the pieces of data according to method of retrieval.
0000   179 ;
0000   180 ;In some cases, the piece of data to be returned by the service requires
0000   181 ;special processing to fetch, calculate, or format it before returning it.
0000   182 ;In these cases, the code of the system service needs to be enhanced.
0000   183 ;If the data returned is a new format for DCL, the lexical function
0000   184 ;module of DCL may need to be enhanced. This is also true for the RTL code.
```

```
                                M 7
0000    186 ;The Macros
0000    187 ;----------
0000    188 ;
0000    189 ;A two-level scheme exists for defining the item tables used by the three
0000    190 ;services and the other facilities. A commonly defined macro (called
0000    191 ;JPI_GENERATE_TABLE, SYI_GENERATE_TABLE, or DVI_GENERATE_TABLE) contains
0000    192 ;multiple calls to a lower-level macro (called JPI_ITEM_CODE, SYI_ITEM_CODE,
0000    193 ;or DVI_ITEM_CODE) which actually defines each element in the table.
0000    194 ;While the _GENERATE_TABLE macros are commonly defined, the _ITEM_CODE macros
0000    195 ;are individually defined according to the needs of facility. (For instance,
0000    196 ;the LEXICON module must store the name of the item as an ASCIC string - in
0000    197 ;order to match it with the string supplied in the F$GETXXI function call;
0000    198 ;the other facilities need not store the item name in text.)
0000    199 ;
0000    200 ;When an item-code must be added, an additional call to the _ITEM_CODE macro
0000    201 ;must be added to the appropriate _GENERATE_TABLE macro. In the case of GETJPI
0000    202 ;and GETDVI, the _GENERATE_TABLE macro is defined in the JPITABLE and DVITABLE
0000    203 ;modules. The SYI_GENERATE_TABLE macro is defined by the SYSPARAM module
0000    204 ;- all the calls to the PARAMETER and PQL macros are 'collected' into the
0000    205 ;SYI_GENERATE_TABLE macro. When used in that mode (when GETSYISW is defined),
0000    206 ;the SYI_ITEMTABLES macro also becomes part of the SYI_GENERATE_TABLE macro.
0000    207 ;SYI_ITEMTABLES is defined in the SYITABLE module and contains all the calls
0000    208 ;to the SYI_ITEM_CODE macro that are Not related to SYSBOOT parameters.
0000    209 ;When GETSYISW is defined in SYSPARAM, the PARAMETER macro does not allocate
0000    210 ;or store memory, but rather passes some of the arguments to it on through via
0000    211 ;a call to SYI_ITEM_CODE. That is how all the calls to PARAMETER become calls
0000    212 ;to SYI_ITEM_CODE.
0000    213 ;
0000    214 ;The following is the situation that exists when the symbol GETSYISW is defined.
0000    215 ;The non-SYSBOOT items are defined by the macro SYI_ITEMTABLES in SYITABLE.MAR.
0000    216 ;The SYSBOOT items are defined by each invokation of the PARAMETER macro in
0000    217 ;SYSPARAM.MAR. Note that each invokation of the PQL macro in SYSPARAM.MAR
0000    218 ;invokes the PARAMETER macro twice. When GETSYISW is defined, the PARAMETER
0000    219 ;macro merely passes its arguments through to a call to the SYI_ITEM_CODE
0000    220 ;macro. The SYI_ITEM_CODE macro is locally defined as needed by the facility.
0000    221 ;
0000    222 ;+-------------------------------------------------------------------------+
0000    223 ;|                        SYI_GENERATE_TABLE                               |
0000    224 ;|   SYI_ITEMTABLES                                                        |
0000    225 ;|                                 PARAMETER          PQL                  |
0000    226 ;|                                                PARAMETER  PARAMETER     |
0000    227 ;|                 |                                                       |
0000    228 ;|SYI_ITEM_CODE|SYI_ITEM_CODE|SYI_ITEM_CODE|SYI_ITEM_CODE|SYI_ITEM_CODE|
0000    229 ;+-------------------------------------------------------------------------+
0000    230 ;\                          / \                                           /
0000    231 ; \                        /   \                                         /
0000    232 ;     FROM SYITABLE.MAR                FROM SYSPARAM.MAR
0000    233 ;     (NON-SYSBOOT ITEMS)               (SYSBOOT ITEMS)
```

```
                     0000   235          .SBTTL  DECLARATIONS
                     0000   236          $ARCDEF                         ; architectural flags
                     0000   237          $CLUBDEF                        ; cluster block definitions
                     0000   238          $CSBDEF                         ; cluster system block definitions
                     0000   239          $IPLDEF                         ; IPL definitions
                     0000   240          $PCBDEF                         ; define processor control block
                     0000   241          $PFLDEF                         ; page file control block
                     0000   242          $PRDEF                          ; define processor registers
                     0000   243          $PSLDEF                         ; define processor status register
                     0000   244          $SBDEF                          ; system block definitions
                     0000   245          $SSDEF                          ; define status codes
                     0000   246          $SYIDEF                         ; define GETSYI item identifiers
                     0000   247          $VADEF                          ; virtual addressing definitions
                     0000   248
                     0000   249 ;
                     0000   250 ; Define the following symbol so that SYSPARAM macros will conditionalize
                     0000   251 ; correctly for us.
                     0000   252 ;
                00000000  0000   253          GETSYISW = 0
                     0000   254
                     0000   255 ;
                     0000   256 ; MACROS:
                     0000   257 ;
                     0000   258
                     0000   259 ;
                     0000   260 ; Macros to define entries in the four item information tables.
                     0000   261 ; There is a table for each data structure from which the user may
                     0000   262 ; request information, and one table for information returned as an
                     0000   263 ; address.  Tables are indexed by low byte of item identifier.
                     0000   264 ; Refer to "OWN STORAGE:" for pictures of the table entries.
                     0000   265 ;
                     0000   266
                     0000   267          .MACRO  SYI_ITEM_CODE   BASE,-          ; for service to use
                     0000   268                                  NAME,-          ; of the item-code
                     0000   269                                  SOURCE,-        ; of the data
                     0000   270                                  DTYPE,-         ; of returned value
                     0000   271                                  BITPOS,-        ; of FLD type data
                     0000   272                                  BITSIZ,-        ;  "
                     0000   273                                  OUTLEN          ; of returned value
                     0000   274
                     0000   275          .IF NOT_DEFINED SYI$_'NAME
                     0000   276
                     0000   277          .IF IDENTICAL <BASE><EXE>
                     0000   278
                     0000   279          .WARN ; SYI$_'NAME IS NOT DEFINED AS 'EXE' IN STARDEFQZ.SDL
                     0000   280
                     0000   281          .ENDC   ; IDENTICAL
                     0000   282
                     0000   283          .IF IDENTICAL <BASE><FLD>
                     0000   284
                     0000   285          .WARN ; SYI$_'NAME IS NOT DEFINED AS 'FLD' IN STARDEFQZ.SDL
                     0000   286
                     0000   287          .ENDC   ; IDENTICAL
                     0000   288
                     0000   289          .ENDC   ; NOT_DEFINED
                     0000   290
                     0000   291          STEP = 5
```

```
0000   292          .IIF IDENTICAL <BASE><EXE>,     STEP = 5
0000   293          .IIF IDENTICAL <BASE><FLD>,     STEP = 7
0000   294
0000   295          XTYPE = VALUE
0000   296          .IIF IDENTICAL <DTYPE><HEXNUM>, XTYPE = VALUE
0000   297          .IIF IDENTICAL <DTYPE><DECNUM>, XTYPE = VALUE
0000   298          .IIF IDENTICAL <DTYPE><PRVMSK>, XTYPE = VALUE
0000   299          .IIF IDENTICAL <DTYPE><PADSTR>, XTYPE = BSTRING
0000   300          .IIF IDENTICAL <DTYPE><HEXSTR>, XTYPE = BSTRING
0000   301          .IIF IDENTICAL <DTYPE><CNTSTR>, XTYPE = CSTRING
0000   302          .IIF IDENTICAL <DTYPE><STRDSC>, XTYPE = VALUE
0000   303          .IIF IDENTICAL <DTYPE><BITVEC>, XTYPE = VALUE
0000   304          .IIF IDENTICAL <DTYPE><BITVAL>, XTYPE = VALUE
0000   305          .IIF IDENTICAL <DTYPE><STDUIC>, XTYPE = VALUE
0000   306          .IIF IDENTICAL <DTYPE><STDTIM>, XTYPE = VALUE
0000   307
0000   308          . = BASE'TBL + <<SYI$_'NAME & ^XFFF> * STEP>
0000   309
0000   310          .IIF    IDENTICAL <BASE><FLD>,  .WORD   <BITSIZ-1>@11!BITPOS
0000   311
0000   312          .LONG   SOURCE
0000   313          .BYTE   XTYPE@5!OUTLEN
0000   314
0000   315          .ENDM   SYI_ITEM_CODE
0000   316
0000   317  ;
0000   318  ; This macro defines the entries to the table of special items.
0000   319  ; The items in this table must be handled by action routines
0000   320  ; before being returned.  Each entry has a word item identifier
0000   321  ; followed by the address of an action routine.
0000   322  ; ALL PROCESSOR REGISTER ITEMS ARE SPECIALS.
0000   323  ;
0000   324
0000   325          .MACRO  SPECIAL_ITEM NAME,ROUTINE
0000   326          .WORD   SYI$_'NAME
0000   327          .ADDRESS ROUTINE
0000   328          .ENDM   SPECIAL_ITEM
0000   329
0000   330  ;
0000   331  ; This macro defines flag bits.
0000   332  ;
0000   333
0000   334          .MACRO  SYIBITS NAME,SIZE
0000   335          SYI_V_'NAME' = SYI_BIT
0000   336          SYI_S_'NAME' = SIZE
0000   337          SYI_BIT = SYI_BIT + SIZE
0000   338          .ENDM   SYIBITS
0000   339
0000   340  ;
0000   341  ; EQUATED SYMBOLS:
0000   342  ;
0000   343
00000004 0000   344          EFN     = 4                     ; event flag number argument
00000008 0000   345          NULLARG1 = 8                    ; first null argument
0000000C 0000   346          NULLARG2 = 12                   ; second null argument
00000010 0000   347          ITMLST  = 16                    ; address of item identifiers
00000014 0000   348          IOSB    = 20                    ; I/O status block address
```

```
00000018  0000   349          ASTADR   = 24                        ; ast routine address
0000001C  0000   350          ASTPRM   = 28                        ; ast parameter
00000002  0000   351          MAXSTRUC = 2                         ; maximum structure code
00000000  0000   352          VALUE    = 0                         ; datatypes
00000001  0000   353          BSTRING  = 1
00000002  0000   354          CSTRING  = 2
FFFFFFE0  0000   355          LOCAL_SPACE = -32                    ; 8 longwords on stack
FFFFFFE0  0000   356          BITSIZ   = LOCAL_SPACE+0
FFFFFFE4  0000   357          BITPOS   = LOCAL_SPACE+4
FFFFFFE8  0000   358          TEMPORARY = LOCAL_SPACE+8
FFFFFFEC  0000   359          SPECIAL_SPACE = LOCAL_SPACE+12
FFFFFFFC  0000   360          FLAGS    = LOCAL_SPACE+28
          0000   361
          0000   362 ;
          0000   363 ; Bit definitions for flags longword on stack
          0000   364 ;
          0000   365
00000000  0000   366          SYI_BIT = 0
          0000   367          $SYIBITS WILD,1                      ; we're doing a wildcard operation
          0000   368          $SYIBITS INCLUSTER,1                 ; we're in a live cluster
          0000   369          $SYIBITS REMOTE_NODE,1               ; the target node isn't the local node
          0000   370          $SYIBITS RETIRED,1                   ; the item-code isn't in use anymore
          0000   371
          0000   372 ;
          0000   373 ; Max structure number definitions
          0000   374 ;
          0000   375
00000101  0000   376          MAX_EXE_ITEM = <SYI$_LASTEXE&^XFFF>-1    ; maximum EXE item number
0000002A  0000   377          MAX_FLD_ITEM = <SYI$_LASTFLD&^XFFF>-1    ; maximum FLD item number
          0000   378
          0000   379 ;
          0000   380 ; OWN STORAGE:
          0000   381 ;
          0000   382
00000000  0000   383          .PSECT  YF$$SYSGETSYI
          0000   384
          0000   385 ;
          0000   386 ; This array contains the maximum item number for each of the two
          0000   387 ; item data structures, indexed by structure number.
          0000   388 ;
          0000   389 MAXCOUNT:
0101      0000   390          .WORD   MAX_EXE_ITEM
002A      0002   391          .WORD   MAX_FLD_ITEM
          0004   392
```

```
                0004   394 ;
                0004   395 ; The following tables are zeroed explicitly to allow the code to
                0004   396 ; recognize an uninitialized element (because of a retired item-code)
                0004   397 ;
                0004   398
                0004   399 EXETBL:
                0004   400         ;-----------------------;
                0004   401         ; .LONG SOURCE          ;
                0004   402         ; .BYTE DTYPE@5!OUTLEN  ;
                0004   403         ;-----------------------;
                0004   404
                0004   405         .REPEAT 5*<MAX_EXE_ITEM+1>
                0004   406         .BYTE   0
 00             0004   407         .ENDR
                050E   408
                050E   409 FLDTBL:
                050E   410         ;---------------------------;
                050E   411         ; .WORD <BITSIZ-1>@11!BITPOS ;
                050E   412         ; .LONG SOURCE               ;
                050E   413         ; .BYTE DTYPE@5!OUTLEN       ;
                050E   414         ;---------------------------;
                050E   415
                050E   416         .REPEAT 7*<MAX_FLD_ITEM+1>
                050E   417         .BYTE   0
 00             050E   418         .ENDR
                063B   419
                063B   420         .SAVE
                063B   421
                063B   422 ;**************************************************************
                063B   423 ;
                063B   424 ; GENERATE THE TABLES USING THE COMMONLY DEFINED MACRO
                063B   425 ;
                063B   426 ;**************************************************************
                063B   427
                063B   428         SYI_GENERATE_TABLE
```

```
                     0243
                     0243                .NLIST  CND
                     0243                PARAMETER       ADDRESS=EXE$GL_DEFFLAGS,-
                     0243                                DEFAULT=1,-
                     0243                                MAX=1,-
                     0243                                MIN=0,-
                     0243                                NAME=BUGREBOOT,-
                     0243                                BIT=EXE$V_BUGREBOOT,-
                     0243                                TYPE=<DYNAMIC,SYS>,-
                     0243                                UNIT=Boolean
00000004             0243                OUTLEN = 4
                     0243                SYI_ITEM_CODE   FLD,-
                     0243                                <BUGREBOOT>,-
                     0243                                <EXE$GL_DEFFLAGS>,-
                     0243                                BITVAL,-
                     0243                                <EXE$V_BUGREBOOT>,-
                     0243                                1,-
                     0243                                1
                     0243
                     0243
00000005             0243                STEP = 5
                     0243
00000000             0243                XTYPE = VALUE
                     0243
00000515             0243                . = FLDTBL + <<SYI$_BUGREBOOT & ^XFFF> * STEP>
                     0515
                     0517
00000000'            0517                .LONG   EXE$GL_DEFFLAGS
      01             051B                .BYTE   XTYPE@5!1
                     051C
                     051C
                     051C
                     051C
```

```
                03F6   429
0000063B        063B   430          .RESTORE
                063B   431
                063B   432 ;
                063B   433 ; Table to define items which must be handled by action routines
                063B   434 ;
                063B   435
                063B   436 SPECIAL:
                063B   437          SPECIAL_ITEM    CLUSTER_MEMBER, SPC_MEMBER
                0641   438          SPECIAL_ITEM    CLUSTER_NODES, SPC_CLUB
                0647   439          SPECIAL_ITEM    CLUSTER_VOTES, SPC_CLUB
                064D   440          SPECIAL_ITEM    CLUSTER_QUORUM, SPC_CLUB
                0653   441          SPECIAL_ITEM    CLUSTER_FSYSID, SPC_CLUB
                0659   442          SPECIAL_ITEM    CLUSTER_FTIME, SPC_CLUB
                065F   443          SPECIAL_ITEM    NODE_CSID, SPC_CSB
                0665   444          SPECIAL_ITEM    NODE_VOTES, SPC_CSB
                066B   445          SPECIAL_ITEM    NODE_QUORUM, SPC_CSB
                0671   446          SPECIAL_ITEM    NODE_SYSTEMID, SPC_SB
                0677   447          SPECIAL_ITEM    NODE_AREA, SPC_SB
                067D   448          SPECIAL_ITEM    NODE_NUMBER, SPC_SB
                0683   449          SPECIAL_ITEM    NODE_SWINCARN, SPC_SB
                0689   450          SPECIAL_ITEM    NODE_SWTYPE, SPC_SB
                068F   451          SPECIAL_ITEM    NODE_SWVERS, SPC_SB
                0695   452          SPECIAL_ITEM    NODE_HWTYPE, SPC_SB
                069B   453          SPECIAL_ITEM    NODE_HWVERS, SPC_SB
                06A1   454          SPECIAL_ITEM    NODENAME, SPC_SB
                06A7   455          SPECIAL_ITEM    SCS_EXISTS, SPC_EXISTS
                06AD   456          SPECIAL_ITEM    SID, SPC_PROCREG
                06B3   457          SPECIAL_ITEM    CPU, SPC_PROCREG
                06B9   458          SPECIAL_ITEM    PAGEFILE_PAGE, SPC_PAGESWAP
                06BF   459          SPECIAL_ITEM    SWAPFILE_PAGE, SPC_PAGESWAP
                06C5   460          SPECIAL_ITEM    PAGEFILE_FREE, SPC_PAGESWAP
                06CB   461          SPECIAL_ITEM    SWAPFILE_FREE, SPC_PAGESWAP
                06D1   462          SPECIAL_ITEM    QUANTUM, SPC_NEGATIVE
0000001A        06D7   463 SPECIAL_LEN = <.-SPECIAL>/6
                06D7   464
```

```
06D7   466          .SBTTL  SYSGETSYI - GETSYI main program
06D7   467
06D7   468 ;++
06D7   469 ;
06D7   470 ; FUNCTIONAL DESCRIPTION:
06D7   471 ;
06D7   472 ;       This service allows a process to receive status and identification
06D7   473 ;       information about the system on which the calling process is running.
06D7   474 ;
06D7   475 ; CALLING SEQUENCE:
06D7   476 ;
06D7   477 ;       CALLS/CALLG
06D7   478 ;
06D7   479 ; INPUTS:
06D7   480 ;
06D7   481 ;       EFN(AP) = number of the event flag to set when all of the requested
06D7   482 ;                 data is valid.
06D7   483 ;       NODE(AP) = pointer to nodename descriptor
06D7   484 ;       CSIDADR(AP) = address of CSID source/destination
06D7   485 ;       ITMLST(AP) = address of a list of item descriptors of the form:
06D7   486 ;
06D7   487 ;                  +------------------------------+
06D7   488 ;                  !  ITEM CODE   !  BUF. LENGTH !
06D7   489 ;                  +------------------------------+
06D7   490 ;                  !      BUFFER ADDRESS          !
06D7   491 ;                  +------------------------------+
06D7   492 ;                  !  ADDRESS TO RETURN LENGTH    !
06D7   493 ;                  +------------------------------+
06D7   494 ;
06D7   495 ;       IOSB(AP) = address of a quadword I/O status block to receive final
06D7   496 ;                  status
06D7   497 ;       ASTADR(AP) = address of an AST routine to be called when all of the
06D7   498 ;                  requested data has been supplied.
06D7   499 ;       ASTPRM(AP) = 32 bit ast parameter
06D7   500 ;
06D7   501 ; IMPLICIT INPUTS:
06D7   502 ;
06D7   503 ;       none
06D7   504 ;
06D7   505 ; OUTPUTS:
06D7   506 ;
06D7   507 ;       none
06D7   508 ;
06D7   509 ; IMPLICIT OUTPUTS:
06D7   510 ;
06D7   511 ;       none
06D7   512 ;
06D7   513 ; ROUTINE VALUE:
06D7   514 ;
06D7   515 ;       SS$_NORMAL -> normal completion
06D7   516 ;       SS$_EXASTLM -> AST quota exceeded
06D7   517 ;       SS$_ACCVIO -> ITMLST can not be read by the calling access mode,
06D7   518 ;                     or the return buffer or return length word can not
06D7   519 ;                     be written by the calling access mode
06D7   520 ;       SS$_BADPARAM -> an invalid item identifier was supplied
06D7   521 ;
06D7   522 ; SIDE EFFECTS:
```

```
                                                H 8
                     06D7   523 ;
                     06D7   524 ;       none
                     06D7   525 ;--
                     06D7   526
                00000000    527         .PSECT  YEXEPAGED       ; only entry mask in this program section
                     0000   528
                   0FFC 0000   529         .ENTRY  EXE$GETSYI,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          06D2'     31 0002   530         BRW     EXE_GETSYI      ; transfer to real procedure
                     0005   531
                000006D7    532         .PSECT  YF$$SYSGETSYI
                     06D7   533
                     06D7   534 ;
                     06D7   535 ; Allocate some local space on the stack
                     06D7   536 ;
                     06D7   537 EXE_GETSYI:
5E      E0 AE     DE 06D7   538         MOVAL   LOCAL_SPACE(SP), SP
        FC AD     D4 06DB   539         CLRL    FLAGS(FP)               ; reset the flags longword
          030D    30 06DE   540         BSBW    NAMCSID                 ; process nodename/CSID pair
        18 50     E9 06E1   541         BLBC    R0,4$
                     06E4   542
                     06E4   543 ;
                     06E4   544 ; Check for and clear possible IOSB
                     06E4   545 ;
51      14 AC     D0 06E4   546         MOVL    IOSB(AP),R1             ; get IOSB address
             08    13 06E8   547         BEQL    3$                      ; branch if none
                     06EA   548         IFNOWRT #8,(R1),30$             ; check write access to it
             61    7C 06F0   549         CLRQ    (R1)                    ; clear IOSB
                     06F2   550 ;
                     06F2   551 ; Check for and clear event flag
                     06F2   552 ;
     53      04 AC    9A 06F2   553 3$:     MOVZBL  EFN(AP),R3              ; get event flag number
00000000'GF          16 06F6   554         JSB     G^SCH$CLREF             ; clear event flag
           6B 50     E9 06FC   555 4$:     BLBC    R0,GRET                 ; and return on errors
                     06FF   556 ;
                     06FF   557 ; Validate AST, if present
                     06FF   558 ;
           18 AC     D5 06FF   559         TSTL    ASTADR(AP)
              0C     13 0702   560         BEQL    5$                      ; no AST to check
54   00000000'GF     D0 0704   561         MOVL    G^CTL$GL_PCB,R4         ; get our PCB address
              38 A4  B5 070B   562         TSTW    PCB$W_ASTCNT(R4)        ; is AST quota exceeded?
                 4B  15 070E   563         BLEQ    40$                     ; branch if so and return error
                     0710   564 ;
                     0710   565 ; Loop through item descriptors, validating the requested item
                     0710   566 ; identifiers and moving accessible items. A zero item identifier
                     0710   567 ; terminates the list.
                     0710   568 ;
     55      10 AC     D0 0710   569 5$:     MOVL    ITMLST(AP),R5           ; get item descriptor list address
                 4C  13 0714   570         BEQL    50$                     ; ITMLST not optional
                     0716   571         IFNORD  #4,(R5),30$             ; check first longword readable
                     071C   572 10$:                                     ; top of item-get loop
        56     85    3C 071C   573         MOVZWL  (R5)+,R6                ; get buffer size
        51     85    3C 071F   574         MOVZWL  (R5)+,R1                ; get item identifier
                 43  13 0722   575         BEQL    60$                     ; done if zero, take normal exit
                     0724   576         IFNORD  #12,(R5),30$            ; check rest of this descriptor ...
                     072A   577                                         ; plus first longword of next one
        57     85    7D 072A   578         MOVQ    (R5)+,R7                ; R7 = buffer address, R8 = length address
                 51  DD 072D   579         PUSHL   R1                      ; save R1 across accessibility check
```

```
                50    57    D0  072F   580          MOVL    R7,R0                   ; buffer address to R0
                51    56    D0  0732   581          MOVL    R6,R1                   ; and size to R1
                      53    D4  0735   582          CLRL    R3                      ; PROBE will use PSL<PRVMOD>
          00000000'EF   16    D0  0737   583          JSB     EXE$PROBEW              ; check write accessibility of buffer
                16 50   E9    D0  073D   584          BLBC    R0,30$                  ; buffer not accessible
                   51 8ED0    D0  0740   585          POPL    R1                      ; restore R1 for use by CHECKITEM
                      55    DD  0743   586          PUSHL   R5                      ; save R5 across item check
                 0075   30    D0  0745   587          BSBW    CHECKITEM               ; check item's validity
                17 50   E9    D0  0748   588          BLBC    R0,50$                  ; return error if not valid
                 00F8   30    D0  074B   589          BSBW    PUTDATA                 ; put the item requested in user buffer
                   55 8ED0    D0  074E   590          POPL    R5                      ; unsave R5
                C8 50   E8    D0  0751   591          BLBS    R0,10$                  ; continue on success
                   14   11    D0  0754   592          BRB     GRET
                                  0756   593
                                  0756   594 ;
                                  0756   595 ; Error/success dispatch points:
                                  0756   596 ;
                50    0C    3C  0756   597 30$:     MOVZWL  #SS$_ACCVIO,R0          ; access violation
                      0F    11  0759   598          BRB     GRET                    ; terminate service below
          50    2A04  8F    3C  075B   599 40$:     MOVZWL  #SS$_EXASTLM,R0         ; AST quota exceeded
                      08    11  0760   600          BRB     GRET                    ; terminate service below
                50    14    3C  0762   601 50$:     MOVZWL  #SS$_BADPARAM,R0        ; illegal item or request
                      03    11  0765   602          BRB     GRET                    ; terminate service below
                50    01    3C  0767   603 60$:     MOVZWL  #SS$_NORMAL,R0          ; normal return
                                  076A   604 ;
                                  076A   605 ; Set the event flag, post completion status, and declare completion AST
                                  076A   606 ;
                      50    DD  076A   607 GRET:    PUSHL   R0                      ; save completion status
          54  00000000'GF   D0  076C   608          MOVL    G^CTL$GL_PCB,R4         ; get PCB address
                51    60 A4   D0  0773   609          MOVL    PCB$L_PID(R4),R1        ; get process's PID
                      52    D4  0777   610          CLRL    R2                      ; set null priority increment
                53    04 AC   9A  0779   611          MOVZBL  EFN(AP),R3              ; get event flag number to set
                00000000'GF   16  077D   612          JSB     G^SCH$POSTEF            ; set the event flag
                51    14 AC   D0  0783   613 10$:     MOVL    IOSB(AP),R1             ; get address of IOSB
                      09    13  0787   614          BEQL    20$                     ; branch if none
                                  0789   615          IFNOWRT #8,(R1),20$             ; check if writable
                   61    6E   D0  078F   616          MOVL    (SP),(R1)               ; store completion status
                55    18 AC   D0  0792   617 20$:     MOVL    ASTADR(AP),R5           ; get address of AST routine
                      15    13  0796   618          BEQL    30$                     ; branch if none specified
                      54    DC  0798   619          MOVPSL  R4                      ; get PSL
          54    54    02    16  EF  079A   620          EXTZV   #PSL$V_PRVMOD,#PSL$S_PRVMOD,R4,R4 ; extract previous mode
                                  079F   621          $DCLAST_S (R5),ASTPRM(AP),R4    ; queue the completion AST
                      50 8ED0    D0  07AD   622 30$:     POPL    R0                      ; restore completion status
                            04    05  07B0   623          RET                     ; and return.
                                  07B1   624
```

```
07B1    626          .SBTTL  CHECKITEM - Validate item identifier
07B1    627
07B1    628 ;++
07B1    629 ;
07B1    630 ; FUNCTIONAL DESCRIPTION:
07B1    631 ;
07B1    632 ;       Routine to validate item identifier and return information
07B1    633 ;       about the item.
07B1    634 ;
07B1    635 ; CALLING SEQUENCE:
07B1    636 ;
07B1    637 ;       JSB/BSB
07B1    638 ;
07B1    639 ; INPUTS:
07B1    640 ;
07B1    641 ;       R1 = item identifier
07B1    642 ;
07B1    643 ; IMPLICIT INPUTS:
07B1    644 ;
07B1    645 ;       none
07B1    646 ;
07B1    647 ; OUTPUTS:
07B1    648 ;
07B1    649 ;       R1 = item identifier
07B1    650 ;       R2 = structure number
07B1    651 ;       R3 = item length
07B1    652 ;       R4 = item source address
07B1    653 ;       R5 = item type code
07B1    654 ;       BITSIZ(FP) - if FLD
07B1    655 ;       BITPOS(FP) - if FLD
07B1    656 ;
07B1    657 ; IMPLICIT OUTPUTS:
07B1    658 ;
07B1    659 ;       none
07B1    660 ;
07B1    661 ; ROUTINE VALUE:
07B1    662 ;
07B1    663 ;       R0 low bit set -> successful return
07B1    664 ;       R0 low bit clear -> invalid item identifier
07B1    665 ;
07B1    666 ; SIDE EFFECTS:
07B1    667 ;
07B1    668 ;       none
07B1    669 ;--
```

```
                     07B1    671 ;
                     07B1    672 ; This table is used to convert the pre V4 GETSYI item-codes to the
                     07B1    673 ; new ones, which have a different form.
                     07B1    674 ;
                     07B1    675 ; Old form:              +--------+--------+
                     07B1    676 ;                        ; 8 bits ; 8 bits ;
                     07B1    677 ;                        +--------+--------+
                     07B1    678 ; SYI$_OLDVERSION =             01       00
                     07B1    679 ; SYI$_OLDCPU =                 02       00
                     07B1    680 ; SYI$_OLDSID =                 02       01
                     07B1    681 ;
                     07B1    682 ; New form:              +---+-------------+
                     07B1    683 ;                        ; 4 ; 12 bits     ;
                     07B1    684 ;                        +---+-------------+
                     07B1    685 ; compatibile with old =   0
                     07B1    686 ; EXE items =              1
                     07B1    687 ; FLD items =              2
                     07B1    688 ;
                     07B1    689
                     07B1    690 COMPAT:
          1000 0100  07B1    691         .WORD   SYI$_OLDVERSION,        SYI$_VERSION
          2000 0200  07B5    692         .WORD   SYI$_OLDCPU,            SYI$_CPU
          1001 0201  07B9    693         .WORD   SYI$_OLDSID,            SYI$_SID
                     07BD    694
                     07BD    695         .ENABLE LOCAL_BLOCK
                     07BD    696
                     07BD    697 CHECKITEM:
               50 D4 07BD    698         CLRL    R0                      ; assume error
     51   F000 8F B3 07BF    699         BITW    #^XF000, R1             ; is it a new item-code?
               18 12 07C4    700         BNEQU   10$                     ; NEQU means it is
          53   03 9A 07C6    701         MOVZBL  #3, R3                  ; setup to scan table
        52   E3 AF 3E 07C9    702         MOVAW   COMPAT-2, R2
                  82 B5 07CD    703 5$:     TSTW    (R2)+               ; skip past new item-code
                  51 82 B1 07CF    704         CMPW    (R2)+, R1           ; does it match this old item-codes?
                     05 12 07D2    705         BNEQU   7$                  ; NEQU means it does not
                  51 62 B0 07D4    706         MOVW    (R2), R1            ; match, use the new itemcode instead
                     05 11 07D7    707         BRB     10$                 ; continue like nothing happened
                     F1 53 F5 07D9    708 7$:     SOBGTR  R3, 5$          ; cycle through the table
                     55 11 07DC    709         BRB     900$                ; error if it wasn't in the table
          52   51 04 0C EF 07DE    710 10$:    EXTZV   #12,#4,R1,R2        ; get the structure number
          53   51 0C 00 EF 07E3    711         EXTZV   #0,#12,R1,R3        ; get the item number
                  02 52 91 07E8    712         CMPB    R2, #MAXSTRUC       ; is it a legal structure number?
                     46 1A 07EB    713         BGTRU   900$                ; GTRU means it is not
             F80B CF42 53 B1 07ED    714         CMPW    R3, MAXCOUNT-2[R2]  ; is it a legal item number?
                     3E 1A 07F3    715         BGTRU   900$                ; GTRU means it is not
                        07F5    716         CASE    R2,<EXE$,FLD$>B,#1  ; goto the appropriate code
                        07FD    717
                  53 05 C4 07FD    718 EXE$:   MULL    #5, R3              ; calc total offset
             53   F7FF CF43 9E 0800    719         MOVAB   EXETBL[R3], R3      ; get address of table element
                     1A 11 0806    720         BRB     50$
                        0808    721
                  53 07 C4 0808    722 FLD$:   MULL    #7, R3              ; calc total offset
             53   FCFE CF43 9E 080B    723         MOVAB   FLDTBL[R3], R3      ; get address of table element
     E0 AD   63   05 0B EF 0811    724         EXTZV   #11,#5,(R3),BITSIZ(FP)  ; get (bitsiz-1) value
                E0 AD D6 0817    725         INCL    BITSIZ(FP)          ; restore its original value
     E4 AD   63   0B 00 EF 081A    726         EXTZV   #0,#11,(R3),BITPOS(FP)  ; get bitpos value
                     83 B5 0820    727         TSTW    (R3)+               ; point to next longword
```

```
                                0822   728
                        54  83  D0  0822   729 50$:    MOVL    (R3)+, R4                       ; get source address
                            0D  13  0825   730          BEQLU   100$                            ; NULL SOURCE MEANS RETIRED ITEM-CODE!!
                                    0827   731                                                 ; IT ALSO MEANS PR$_KSP WILL NEVER BE
                                    0827   732                                                 ; ABLE TO BE AN ITEM-CODE!
                55  63  03  05  EF  0827   733          EXTZV   #5,#3,(R3),R5                   ; get DTYPE
                53  63  05  00  EF  082C   734          EXTZV   #0,#5,(R3),R3                   ; get OUTLEN
                            50  D6  0831   735 70$:     INCL    R0                              ; success!
                                    0833   736
                            05  0833   737 900$:    RSB                                     ; return to caller
                                    0834   738
                    FC  AD  08  C8  0834   739 100$:    BISL    #<1@SYI_V_RETIRED>,FLAGS(FP) ; mark it as obsolete
                        53  04  D0  0838   740          MOVL    #4,R3                           ; src length
                    54  EC  AD  DE  083B   741          MOVAL   SPECIAL_SPACE(FP),R4            ; scratch area
                            64  D4  083F   742          CLRL    (R4)                            ; null answer now
                        55  00  D0  0841   743          MOVL    #VALUE,R5                       ; dtype
                            EB  11  0844   744          BRB     70$                             ; success exit
                                    0846   745
                                    0846   746          .DISABLE LOCAL_BLOCK
```

```
0846   748          .SBTTL  PUTDATA - Put requested data in user buffer
0846   749
0846   750 ;++
0846   751 ;
0846   752 ; FUNCTIONAL DESCRIPTION:
0846   753 ;
0846   754 ;       This routine moves the requested data to the user's buffer and
0846   755 ;       returns the actual data length to the user. It assumes that the
0846   756 ;       user's buffer has been probed.
0846   757 ;
0846   758 ; CALLING SEQUENCE:
0846   759 ;
0846   760 ;       JSB/BSB
0846   761 ;
0846   762 ; INPUTS:
0846   763 ;
0846   764 ;       R1 = item identifier
0846   765 ;       R2 = data structure number
0846   766 ;       R3 = item length
0846   767 ;       R4 = item address
0846   768 ;       R5 = item type code
0846   769 ;       R6 = user buffer length
0846   770 ;       R7 = user buffer address
0846   771 ;       R8 = address to return length
0846   772 ;       BITSIZ(FP)
0846   773 ;       BITPOS(FP)
0846   774 ;
0846   775 ; IMPLICIT INPUTS:
0846   776 ;
0846   777 ;       none
0846   778 ;
0846   779 ; OUTPUTS:
0846   780 ;
0846   781 ;       none
0846   782 ;
0846   783 ; IMPLICIT OUTPUTS:
0846   784 ;
0846   785 ;       none
0846   786 ;
0846   787 ; ROUTINE VALUE:
0846   788 ;
0846   789 ;       R0 low bit set -> success
0846   790 ;       R0 low bit clear -> access violation on write of length
0846   791 ;
0846   792 ; SIDE EFFECTS:
0846   793 ;
0846   794 ;       Registers R1-R4 destroyed
0846   795 ;--
```

```
                                0846   797
                                0846   798 PUTDATA:
                                0846   799
                                0846   800 ;
                                0846   801 ; Call routine to check for special conditions
                                0846   802 ;
                                0846   803
            27 FC AD    03   E0  0846   804          BBS     #SYI_V_RETIRED,FLAGS(FP),15$ ; skip alot for oldie items
                        53   10  084B   805          BSBB    CHECK_SPC                    ; handle special items
                     4F 50   E9  084D   806          BLBC    R0,50$                       ; were we successful?
                                0850   807
                                0850   808 ;
                                0850   809 ; Check for counted string, and find actual length if so.
                                0850   810 ;
                                0850   811
                     55 02   D1  0850   812          CMPL    #CSTRING,R5                  ; is this special string?
                        03   12  0853   813          BNEQ    10$                          ; branch if not
                     53 84   9A  0855   814          MOVZBL  (R4)+,R3                     ; get length and skip length byte
                                0858   815 ;
                                0858   816 ; Fetch the bitpos and bitsiz if it's a FLD item
                                0858   817 ;
                     02 52   D1  0858   818 10$:     CMPL    R2, #SYI$C_FLDTYPE           ; is it FLD?
                        15   12  085B   819          BNEQU   15$                          ; NEQU means it is not
                  50    E0 AD   D0  085D   820          MOVL    BITSIZ(FP), R0               ; get bitsiz
                  51    E4 AD   D0  0861   821          MOVL    BITPOS(FP), R1               ; get bitpos
                  E8 AD    64   D0  0865   822          MOVL    (R4), TEMPORARY(FP)          ; get copy of cell
                  54    E8 AD   DE  0869   823          MOVAL   TEMPORARY(FP), R4            ; point to copy
            64    64    50   51   EF  086D   824          EXTZV   R1,R0,(R4),(R4)              ; fetch sub-field and save
                                0872   825 ;
                                0872   826 ; Move the data
                                0872   827 ;
                        28   BB  0872   828 15$:     PUSHR   #^M<R3,R5>                   ; save needed registers from movc
      67    56    00    64    53   2C  0874   829          MOVC5   R3,(R4),#0,R6,(R7)           ; move data to user's buffer, zero fill
                        28   BA  087A   830          POPR    #^M<R3,R5>                   ; restore registers
                        58   D5  087C   831          TSTL    R8                           ; did caller want return length?
                        18   13  087E   832          BEQL    30$                          ; branch if not
                                0880   833          IFNOWRT #2,(R8),40$                  ; exit if word not writable
                  02 FC AD    03   E1  0886   834          BBC     #SYI_V_RETIRED,FLAGS(FP),18$
                        53   D4  088B   835          CLRL    R3                           ; make his retlen null
                     56 53   D1  088D   836 18$:     CMPL    R3,R6                        ; see how much was moved
                        03   15  0890   837          BLEQ    20$                          ; use valid data length if it fit
                     53 56   D0  0892   838          MOVL    R6,R3                        ; else give him "too short" buffer size
                     68 53   B0  0895   839 20$:     MOVW    R3,(R8)                      ; return length to user
                                0898   840
                     50 01   3C  0898   841 30$:     MOVZWL  #SS$_NORMAL,R0               ; set success code
                        05   RSB 089B   842
                                089C   843
                     50 0C   3C  089C   844 40$:     MOVZWL  #SS$_ACCVIO,R0               ; set error code
                        05   RSB 089F   845 50$:     RSB                                  ; return
```

```
08A0    847          .SBTTL  SPECIAL - Handle special conditions
08A0    848
08A0    849 ;++
08A0    850 ;
08A0    851 ; FUNCTIONAL DESCRIPTION:
08A0    852 ;
08A0    853 ;       These routines handle data items which must be transformed
08A0    854 ;       before they are returned to the user.  Generally, some
08A0    855 ;       transformation is applied to the data item and the newly
08A0    856 ;       computed item is stored in SPECIAL_SPACE on the stack.
08A0    857 ;       The handling routine then changes R4 to point to SPECIAL_SPACE
08A0    858 ;       so that PUTDATA will move the item from local storage.
08A0    859 ;
08A0    860 ; CALLING SEQUENCE:
08A0    861 ;
08A0    862 ;       JSB/BSB
08A0    863 ;
08A0    864 ; INPUTS:
08A0    865 ;
08A0    866 ;       R1 = item identifier
08A0    867 ;       R3 = item length
08A0    868 ;       R4 = item address/offset
08A0    869 ;       R9 = target CSB address
08A0    870 ;       R11 = target CSID
08A0    871 ;
08A0    872 ; IMPLICIT INPUTS:
08A0    873 ;
08A0    874 ;       none
08A0    875 ;
08A0    876 ; OUTPUTS:
08A0    877 ;
08A0    878 ;       none
08A0    879 ;
08A0    880 ; IMPLICIT OUTPUTS:
08A0    881 ;
08A0    882 ;       none
08A0    883 ;
08A0    884 ; ROUTINE VALUE:
08A0    885 ;
08A0    886 ;       none
08A0    887 ;
08A0    888 ; SIDE EFFECTS:
08A0    889 ;
08A0    890 ;       none
08A0    891 ;--
```

```
                        08A0   893
                        08A0   894 CHECK_SPC:
                        08A0   895
                        08A0   896 ;
                        08A0   897 ; Registers R5 - R8 are saved at this level and may be used by
                        08A0   898 ; the action routines without being saved.  Action routines are JSB'ed
                        08A0   899 ; to with R5 containing the address of SPECIAL_SPACE on the stack.
                        08A0   900 ;
                        08A0   901
      01E0 8F    BB     08A0   902          PUSHR   #^M<R5,R6,R7,R8>        ; save resisters
 55     EC AD    DE     08A4   903          MOVAL   SPECIAL_SPACE(FP),R5    ; local storage for action routine
           65    7C     08A8   904          CLRQ    (R5)                    ; clear the special buffer
        08 A5    7C     08AA   905          CLRQ    8(R5)
        50 01    3C     08AD   906          MOVZWL  #SS$_NORMAL,R0          ; assume success
        57 1A    D0     08B0   907          MOVL    #SPECIAL_LEN,R7         ; get number of table entries
 58   FD84 CF    DE     08B3   908          MOVAL   SPECIAL,R8              ; get address of table
                        08B8   909 10$:
        88 51    B1     08B8   910          CMPW    R1,(R8)+                ; does entry match item?
           13    13     08BB   911          BEQL    20$                     ; yes, go handle it
        58 04    C0     08BD   912          ADDL    #4,R8                   ; skip handler address
          F5 57  F5     08C0   913          SOBGTR  R7,10$                  ; scan rest of table
                        08C3   914
03 FC AD   02    E1     08C3   915          BBC     #SYI_V_REMOTE_NODE,FLAGS(FP),35$ ; nonlocal noncluster info?
        54 55    D0     08C8   916          MOVL    R5,R4                   ; make the returned data null
      01E0 8F    BA     08CB   917 35$:     POPR    #^M<R5,R6,R7,R8>        ; restore registers
                 05     08CF   918          RSB
                        08D0   919
                        08D0   920 20$:
           98    16     08D0   921          JSB     @(R8)+                  ; call action routine
           F7    11     08D2   922          BRB     35$
                        08D4   923
                        08D4   924 ;
                        08D4   925 ; Data handling routines
                        08D4   926 ;*****************************************************************
                        08D4   927 ; ALL NON-CLUSTER SPECIAL DATA ITEMS SHOULD TEST REMOTE_NODE AS BELOW
                        08D4   928 ;*****************************************************************
                        08D4   929 ;
                        08D4   930
                        08D4   931 ;
                        08D4   932 ; Is the SCS code loaded?
                        08D4   933 ;
                        08D4   934
                        08D4   935 SPC_EXISTS:
5C FC AD   02    E0     08D4   936          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
  00000000'EF    D5     08D9   937          TSTL    SCS$GA_EXISTS           ; is the cell empty?
           54    13     08DF   938          BEQLU   POINT_R4                ; null cell means it doesn't exist
           65    D6     08E1   939          INCL    (R5)                    ; make result TRUE
           50    11     08E3   940          BRB     POINT_R4
                        08E5   941
                        08E5   942 ;
                        08E5   943 ; Processor registers require special instructions to fetch
                        08E5   944 ;
                        08E5   945
                        08E5   946 SPC_PROCREG:
4B FC AD   02    E0     08E5   947          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
        65 54    DB     08EA   948          MFPR    R4,(R5)                 ; get the register contents
           46    11     08ED   949          BRB     POINT_R4
```

```
                                08EF   950
                                08EF   951 ;
                                08EF   952 ; Cell is stored as a negative, reverse it and return
                                08EF   953 ;
                                08EF   954
                                08EF   955 SPC_NEGATIVE:
          41 FC AD    02    E0  08EF   956          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
                65    64    CE  08F4   957          MNEGL   (R4), (R5)              ; Negate it
                      3C    11  08F7   958          BRB     POINT_R4
                                08F9   959
                                08F9   960 ;
                                08F9   961 ; This cluster item makes sense even when not in a cluster
                                08F9   962 ;
                                08F9   963
                                08F9   964 SPC_MEMBER:
       65   FC AD    01    01  EF  08F9   965          EXTZV   #SYI_V_INCLUSTER,#1,FLAGS(FP),(R5) ; get the flag
                      34    11  08FF   966          BRB     POINT_R4
                                0901   967
                                0901   968 ;
                                0901   969 ; These are cluster only items
                                0901   970 ;
                                0901   971
                                0901   972 SPC_CLUB:
          2F FC AD    01    E1  0901   973          BBC     #SYI_V_INCLUSTER,FLAGS(FP),POINT_R4 ; return null if no cluster
       57   00000000'EF    D0  0906   974          MOVL    CLU$GL_CLUB,R7          ; get the address of the CLUB
                54  6744    9E  090D   975          MOVAB   (R7)[R4],R4             ; get the address of the field
                      44    11  0911   976          BRB     FETCH_CLU
                                0913   977
                                0913   978 SPC_CSB:
          1D FC AD    01    E1  0913   979          BBC     #SYI_V_INCLUSTER,FLAGS(FP),POINT_R4 ; return null if no cluster
                            00000918  0918   980 SPC_LOCK =
                                0918   981          SETIPL  LOCK                    ; lock the database
                      43    10  091F   982          BSBB    VERIFY_CSB              ; double check the CSB address
                   11 50    E9  0921   983          BLBC    R0,POINT_R4             ; exit if bad
                54  6944    9E  0924   984          MOVAB   (R9)[R4],R4             ; get actual address of field
                      2D    11  0928   985          BRB     FETCH_CLU
                                092A   986
                                092A   987 LOCAL_SB:
                      51    DD  092A   988          PUSHL   R1                      ; save R1
       51   00000000'GF    DE  092C   989          MOVAL   G^SCS$GA_LOCALSB,R1     ; point to our local system block
                      1B    11  0933   990          BRB     GET_SB_FLD              ; go get desired information from SB
                                0935   991
                                0935   992 POINT_R4:
                54    55    D0  0935   993          MOVL    R5,R4                   ; make R4 point to data
                            05  0938   994          RSB
                                0939   995
                                0939   996 SPC_SB:
       EC FC AD    01    E1  0939   997          BBC     #SYI_V_INCLUSTER,FLAGS(FP),LOCAL_SB ; use local SB if no cluster
                                093E   998          SETIPL  LOCK                    ; lock the database
                      1D    10  0945   999          BSBB    VERIFY_CSB              ; double check the CSB address
                   EB 50    E9  0947  1000          BLBC    R0,POINT_R4             ; exit if bad
                      51    DD  094A  1001          PUSHL   R1                      ; save R1
          51   68 A9    D0  094C  1002          MOVL    CSB$L_SB(R9),R1         ; get SB address
                                0950  1003 GET_SB_FLD:
                54  6144    9E  0950  1004          MOVAB   (R1)[R4],R4             ; get actual address of field
                      51 8ED0  0954  1005          POPL    R1                      ; restore R1
                                0957  1006
```

```
                        0957  1007 FETCH_CLU:
                  3F    BB  0957  1008          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; save the register past the MOVC3
          65      64    53  28  0959  1009          MOVC3   R3,(R4),(R5)            ; get the data into the special buffer
                  3F    BA  095D  1010          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; restore the registers
                            095F  1011          SETIPL  #0                      ; drop IPL again
                  D1    11  0962  1012          BRB     POINT_R4
                            0964  1013
                            0964  1014 VERIFY_CSB:
                  56    5B  32  0964  1015          CVTWL   R11,R6                  ; get the system index
      56  00000000'FF46    D0  0967  1016          MOVL    @CLU$GL_CLUSVEC[R6],R6  ; get the CSB address
                  0F    18  096F  1017          BGEQ    99$                     ; GEQ means it is now unused
                59  56    D1  0971  1018          CMPL    R6,R9                   ; is it the same as ours?
                  0A    12  0974  1019          BNEQ    99$                     ; NEQ means it changed
                5B  4C A6   D1  0976  1020          CMPL    CSB$L_CSID(R6),R11      ; is the CSID the same?
                  04    12  097A  1021          BNEQ    99$                     ; NEQ means it changed
                50  01    3C  097C  1022          MOVZWL  #SS$_NORMAL,R0
                            05  097F  1023          RSB
                            0980  1024
                            0980  1025 99$:       SETIPL  #0                      ; drop IPL
                FC AD   02  CA  0983  1026          BICL2   #<1@SYI_V_INCLUSTER>,FLAGS(FP) ; reset the cluster flag
              50    028C 8F  3C  0987  1027          MOVZWL  #SS$_NOSUCHNODE,R0      ; declare an error
                            05  098C  1028          RSB
                            098D  1029
                            098D  1030 LOCK:
                            08  098D  1031          .BYTE   IPL$_SCS
                            098E  1032          ASSUME  <.-SPC_LOCK> LE 512
                            098E  1033 ;
                            098E  1034 ;
                            098E  1035 ; Return total sizes for all page or swap files
                            098E  1036 ;
                            098E  1037 ; Input bit mask in R4
                            098E  1038 ;       bit 0
                            098E  1039 ;               0 -> page file
                            098E  1040 ;               1 -> swap file
                            098E  1041 ;       bit 1
                            098E  1042 ;               0 -> total space
                            098E  1043 ;               1 -> free space
                            098E  1044 ;       bit 2
                            098E  1045 ;               1 -> keeps it from being null
                            098E  1046 ;
                            098E  1047 ;
                            098E  1048 SPC_PAGESWAP:
          A2 FC AD    02    E0  098E  1049          BBS     #SYI_V_REMOTE_NODE,FLAGS(FP),POINT_R4 ; skip it for remotes
                    10  54    E8  0993  1050          BLBS    R4,10$                  ; swap file
      57    00000000'EF     3C  0996  1051          MOVZWL  SGN$GW_SWPFILCT,R7      ; first page file slot
      58    00000000'EF     D0  099D  1052          MOVL    MMG$GL_MAXPFIDX,R8      ; last one
                    0C    11  09A4  1053          BRB     20$                     ; resume in common code
                            09A6  1054 ;
                57  01    D0  09A6  1055 10$:       MOVL    #1,R7                   ; always the first swap file slot
      58    00000000'EF     3C  09A9  1056          MOVZWL  SGN$GW_SWPFILCT,R8
                    58    D7  09B0  1057          DECL    R8                      ; max index for swap files
                            09B2  1058 ;
                    65    D4  09B2  1059 20$:       CLRL    (R5)                    ; initial count
                58  57    C2  09B4  1060          SUBL    R7,R8                   ; slots to count - 1
                    32    19  09B7  1061          BLSS    60$                     ; none
      57    00000000'FF47   DE  09B9  1062          MOVAL   @MMG$GL_PAGSWPVC[R7],R7 ; first slot
                56  87    D0  09C1  1063 30$:       MOVL    (R7)+,R6                ; address of PFL structure
```

```
00000000'EF    56   D1  09C4  1064          CMPL    R6,MMG$GL_NULLPFL       ; is it in use
               13   13  09CB  1065          BEQL    50$                     ; no
   0E 23 A6    00   E1  09CD  1066          BBC     #PFL$V_INITED,PFL$B_FLAGS(R6),50$ ; not inited
      06 54    01   E0  09D2  1067          BBS     #1,R4,40$               ; count free space
      65   14 A6    C0  09D6  1068          ADDL    PFL$L_BITMAPSIZ(R6),(R5) ; total size / 8
               04   11  09DA  1069          BRB     50$
      65   18 A6    C0  09DC  1070 40$:     ADDL    PFL$L_FREPAGCNT(R6),(R5) ; total free pages
           DE 58    F4  09E0  1071 50$:     SOBGEQ  R8,30$                  ; loop over all files
                        09E3  1072 ;
      04 54    01   E0  09E3  1073          BBS     #1,R4,60$               ; free space - already page count
   65 65    03   78  09E7  1074          ASHL    #3,(R5),(R5)            ; convert byte count to page count
         FF47   31  09EB  1075 60$:     BRW     POINT_R4                ; join common exit code
                        09EE  1076
```

```
                        09EE  1078          .SBTTL  NAMCSID - Get specified node CSID
                        09EE  1079 ;++
                        09EE  1080 ;
                        09EE  1081 ; FUNCTIONAL DESCRIPTION:
                        09EE  1082 ;
                        09EE  1083 ;       Routine to convert a node name to a CSID. If a
                        09EE  1084 ;       valid CSID or node name is specified, the standard conversion
                        09EE  1085 ;       routine EXE$NAMCSID is simply called.  If, however, a CSID that implies
                        09EE  1086 ;       a "wildcard" CSID (-1) is specified, then the next active node is
                        09EE  1087 ;       chosen as the node CSID to pass to EXE$NAMCSID.  EXE$NAMCSID then
                        09EE  1088 ;       returns the node's CSB address.
                        09EE  1089 ;
                        09EE  1090 ; INPUTS:
                        09EE  1091 ;
                        09EE  1092 ;       CSIDADR(AP) = address of specified CSID
                        09EE  1093 ;       NODE(AP) = address of specified process name descriptor
                        09EE  1094 ;
                        09EE  1095 ; OUTPUTS:
                        09EE  1096 ;
                        09EE  1097 ;       R0 = success/failure of operation
                        09EE  1098 ;       R4 = current process PCB address
                        09EE  1099 ;       R9 = specified node CSB address
                        09EE  1100 ;       R11 = specified node CSID
                        09EE  1101 ;       @CSIDADR(AP) = specified node CSID or special "wildcard" context CSID
                        09EE  1102 ;--
                        09EE  1103
                00000008 09EE  1104          CSIDADR  = 8
                0000000C 09EE  1105          NODE     = 12
                        09EE  1106
                        09EE  1107 NAMCSID:
                        09EE  1108          .ENABLE LOCAL_BLOCK
                        09EE  1109
                        09EE  1110 ;
                        09EE  1111 ; MAKE SURE WE'RE IN A CLUSTER HERE
                        09EE  1112 ;
56   00000000'EF   D0   09EE  1113          MOVL    CLU$GL_CLUB,R6          ; GET CLUB ADDRESS
           09     13   09F5  1114          BEQL    1$                      ; IF EQL, NOT IN CLUSTER
   04 1C A6   00   E1   09F7  1115          BBC     #CLUB$V_CLUSTER,CLUB$L_FLAGS(R6),1$ ; IF CLEAR, NOT A CLUSTER
      FC AD   02   C8   09FC  1116          BISL2   #<1@SYI_V_INCLUSTER>,FLAGS(FP) ; mark that we're in a cluster
                        0A00  1117
      56   08 AC   D0   0A00  1118 1$:      MOVL    CSIDADR(AP),R6          ; get CSID address
           4D     13   0A04  1119          BEQL    19$                     ; if eql - none
                        0A06  1120          IFWRT   #4,(R6),2$              ; check access to CSID
           00CE   31   0A0C  1121          BRW     50$
         50   66   D0   0A0F  1122 2$:      MOVL    (R6),R0                 ; get CSID
              3F   13   0A12  1123          BEQL    19$                     ; if eql - none
              79   14   0A14  1124          BGTR    20$                     ; if gtr - standard CSID
                        0A16  1125 ;
                        0A16  1126 ; "Wildcard" type CSID specified
                        0A16  1127 ;
                        0A16  1128
   03 FC AD   01   E0   0A16  1129          BBS     #SYI_V_INCLUSTER,FLAGS(FP),5$ ; are we in a cluster?
           00C4   31   0A1B  1130          BRW     60$                     ; wildcarding without a cluster!
                        0A1E  1131 5$:      SETIPL  80$                     ; lock the cluster database
         55   50   32   0A25  1132          CVTWL   R0,R5                   ; get NIX (Node IndeX) from CSID
      FC AD   01   C8   0A28  1133          BISL2   #<1@SYI_V_WILD>,FLAGS(FP) ; mark wildcarding in effect
              55   B6   0A2C  1134 10$:     INCW    R5                      ; increment NIX
```

```
                00000000'EF    55   B1   0A2E  1135          CMPW    R5,CLU$GW_MAXINDEX      ; is NIX in valid range?
                               03   1F   0A35  1136          BLSSU   11$                     ; if LSSU, yep
                           00A8   31   0A37  1137          BRW     60$                     ; no more nodes
                         55    55   3C   0A3A  1138 11$:     MOVZWL  R5,R5                   ; clear out the top half of R5
                50   00000000'FF45   D0   0A3D  1139          MOVL    @CLU$GL_CLUSVEC[R5],R0  ; get CSB address
                               E5   18   0A45  1140          BGEQ    10$                     ; if GEQ, unused - try next one
                      7E    4C A0   D0   0A47  1141          MOVL    CSB$L_CSID(R0),-(SP)    ; get the CSID
                                     0A4B  1142          SETIPL  #0                      ; lower IPL to touch the argument list
                         66    8E   D0   0A4E  1143          MOVL    (SP)+,(R6)              ; store CSID in argument list
                               3C   11   0A51  1144          BRB     20$
                                    0A53  1145
                                    0A53  1146 ;
                                    0A53  1147 ;  At this point, the CSID argument was defaulted
                                    0A53  1148 ;
                   3C FC AD    01   E0   0A53  1149 19$:     BBS     #SYI_V_INCLUSTER,FLAGS(FP),21$ ; let EXE$NAMCSID do it
                      53    0C AC   D0   0A58  1150          MOVL    NODE(AP),R3             ; get the nodename argument
                               71   13   0A5C  1151          BEQL    75$                     ; it was defaulted too, skip everything
                                    0A5E  1152
                                    0A5E  1153 ;
                                    0A5E  1154 ;  At this point, we're not in a cluster, but a nodename was specified,
                                    0A5E  1155 ;  see if it's the local one, if so succeed.
                                    0A5E  1156 ;
                                    0A5E  1157          IFNORD  #8,(R3),50$             ; probe the descriptor
                         52    63   7D   0A64  1158          MOVQ    (R3),R2                 ; get the nodename descriptor
                         52    52   3C   0A67  1159          MOVZWL  R2,R2                   ; is the length legal?
                               7D   13   0A6A  1160          BEQL    65$                     ; EQL means nope
                         52    0F   B1   0A6C  1161          CMPW    #15,R2                  ; is it too long?
                               78   1F   0A6F  1162          BLSSU   65$                     ; LSSU means too long
                                    0A71  1163          IFNORD  R2,(R3),50$             ; probe the string
                51   00000000'GF   DE   0A77  1164          MOVAL   G^SCS$GA_LOCALSB,R1     ; point at the local SB
                      55    44 A1   9E   0A7E  1165          MOVAB   SB$T_NODENAME(R1),R5    ; get address of nodename
                         85    52   91   0A82  1166          CMPB    R2,(R5)+                ; is it the right length?
                               62   12   0A85  1167          BNEQ    65$                     ; NEQ means no
                      65    63   52   29   0A87  1168          CMPC3   R2,(R3),(R5)            ; is it the same nodename?
                               5C   12   0A8B  1169          BNEQ    65$                     ; NEQ means this is NOT the one
                               40   11   0A8D  1170          BRB     75$                     ; It Is the local nodename, exit
                                    0A8F  1171
                                    0A8F  1172 ;
                                    0A8F  1173 ;  Convert node name to CSID, if specified
                                    0A8F  1174 ;
             4E FC AD    01   E1   0A8F  1175 20$:     BBC     #SYI_V_INCLUSTER,FLAGS(FP),60$ ; specified CSID   no cluster!
                   5C    04   C0   0A94  1176 21$:     ADDL    #4,AP                   ; make CSIDADR top argument
                00000000'EF   16   0A97  1177          JSB     25$                     ; get into nonpaged code
                                    0A9D  1178          .SAVE_PSECT                     ; save current .PSECT context
                                    0A9D  1179
                                    0A9D  1180 ;  The reason for jumping to the nonpaged exec rather than dynamically
                                    0A9D  1181 ;  locking down pageable pages is that EXE$NAMCSID cannot be entered
                                    0A9D  1182 ;  above IPL 2 and the dynamic locking would cause that to happen. The
                                    0A9D  1183 ;  reason that EXE$NAMCSID must be entered at IPL 2 or lower is that it
                                    0A9D  1184 ;  touches the caller's argument list (which contains arguments that
                                    0A9D  1185 ;  could fault) and page faults are not allowed above IPL 2.
                                    0A9D  1186
                           00000000 1187          .PSECT  AEXENONPAGED            ; EXE$NAMCSID returns at IPL$_SYNCH
                00000AF1'EF   16   0000  1188 25$:     JSB     EXE$NAMCSID             ; get CSB address and CSID
                                    0006  1189          SETIPL  #0                      ; restore IPL - CSB is no longer locked
                               05   0009  1190          RSB                             ; go back to paged code
                                    000A  1191
```

```
                            00000A9D             .RESTORE_PSECT                  ; get paged .PSECT context back
            5C      04  C2  0A9D                 SUBL    #4,AP                   ; restore argument pointer
            5B      51  D0  0AA0                 MOVL    R1,R11                  ; save CSID
     07 FC  AD      00  E1  0AA3                 BBC     #SYI_V_WILD,FLAGS(FP),30$ ; "wildcard" type CSID specified?
            66      51  B0  0AA8                 MOVW    R1,(R6)                 ; restore node index context
         02 A6      01  AE  0AAB                 MNEGW   #1,2(R6)                ; set continuation context
                            0AAF  ;
                            0AAF  ; Check CSID address and return
                            0AAF  ;
         20 50          E9  0AAF  30$:           BLBC    R0,40$                  ; branch if error
            59      54  D0  0AB2                 MOVL    R4,R9                   ; save CSB address
   54   00000000'EF     D0  0AB5                 MOVL    CLU$GL_CLUB,R4          ; get address of Cluster Block
                    04  12  0ABC                 BNEQU   32$                     ; NEQU means it's not null
                            0ABE                 BUG_CHECK ICONCLUDAT,FATAL      ; oh oh
      54    10      A4  DE  0AC2  32$:           MOVAL   CLUB$L_LOCAL_CSB(R4),R4 ; get address of local CSB
            59      64  D1  0AC6                 CMPL    (R4),R9                 ; see if local csb = target csb
                    04  13  0ACB                 BEQL    75$                     ; EQL means target = local
        FC  AD      04  C8  0ACB                 BISL2   #<1@SYI_V_REMOTE_NODE>,FLAGS(FP) ; set the remote target flag
            50      01  3C  0ACF  75$:           MOVZWL  #SS$_NORMAL,R0          ; set success
   54   00000000'EF     D0  0AD2  40$:           MOVL    SCH$GL_CURPCB,R4        ; restore current PCB address
                            0AD9                 SETIPL  #0                      ; make sure we can page fault
                        05  0ADC                 RSB
                            0ADD
            50      0C  3C  0ADD  50$:           MOVZWL  #SS$_ACCVIO,R0          ; set access violation
                    F0  11  0AE0                 BRB     40$                     ;
     50   0A00      8F  3C  0AE2  60$:           MOVZWL  #SS$_NOMORENODE,R0      ; set no more nodes
                    E9  11  0AE7                 BRB     40$                     ;
     50   028C      8F  3C  0AE9  65$:           MOVZWL  #SS$_NOSUCHNODE,R0      ; set no such node
                    E2  11  0AEE                 BRB     40$                     ;
                            0AF0
                        08  0AF0  80$:           .BYTE   IPL$_SCS                ; to lock the cluster database
                            0AF1                 ASSUME  <.-5$> LE 512
                            0AF1
                            0AF1                 .DISABLE LOCAL_BLOCK
```

```
                                0AF1  1227          .SBTTL  EXE$NAMCSID - CONVERT NODE NAME TO CSID
                                0AF1  1228 ;++
                                0AF1  1229 ;       EXE$NAMCSID  -  CONVERT NODE NAME TO CSID
                                0AF1  1230 ;
                                0AF1  1231 ; FUNCTIONAL DESCRIPTION:
                                0AF1  1232 ;       EXE$NAMCSID OBTAINS THE PROPER CSID AND CSB ADDRESS FOR A
                                0AF1  1233 ;       STANDARD NODE SERVICE ARGUMENT LIST CONSISTING
                                0AF1  1234 ;       OF A CSID/NODE-NAME PAIR.  THE ABSENCE OF BOTH SELECTS THE
                                0AF1  1235 ;       CURRENT NODE.
                                0AF1  1236 ;
                                0AF1  1237 ;       NOTE THAT THE OPERATION OF THIS ROUTINE ONLY MAKES SENSE IN
                                0AF1  1238 ;       A CLUSTER, THEREFORE A NOSUCHNODE ERROR WILL BE RETURNED IF
                                0AF1  1239 ;       CLU$GL_CLUB = 0 ON ENTRY.
                                0AF1  1240 ;
                                0AF1  1241 ; CALLING SEQUENCE:
                                0AF1  1242 ;       JSB/BSB EXE$NAMCSID
                                0AF1  1243 ;
                                0AF1  1244 ; INPUT PARAMETERS:
                                0AF1  1245 ;       CSID(AP) - ADDRESS OF CSID SOURCE/DESTINATION (CSID)
                                0AF1  1246 ;       NODENAME(AP) - POINTER TO NODE DESCRIPTOR TO CONVERT TO CSID
                                0AF1  1247 ;
                                0AF1  1248 ; IMPLICIT INPUTS:
                                0AF1  1249 ;       @CLU$GL_CLUSVEC - VECTOR OF CSB ADDRESSES
                                0AF1  1250 ;
                                0AF1  1251 ; OUTPUT PARAMETERS:
                                0AF1  1252 ;       R0 - COMPLETION STATUS
                                0AF1  1253 ;       R1 - NODE IDENTIFICATION (CSID) OF NAMED NODE.
                                0AF1  1254 ;       R4 - CSB ADDRESS OF NODE IF MATCH IS FOUND.
                                0AF1  1255 ;       @CSID(AP) - NODE IDENTIFICATION (CSID) OF SELECTED NODE
                                0AF1  1256 ;       IPL - IPL$_SYNCH (IPL UNCHANGED IF SS$_ACCVIO OR SS$_IVLOGNAM)
                                0AF1  1257 ;
                                0AF1  1258 ; COMPLETION CODES:
                                0AF1  1259 ;       SS$_NORMAL  -  NORMAL SUCCESSFUL COMPLETION
                                0AF1  1260 ;       SS$_IVLOGNAM  -  INVALID LOGICAL NAME STRING
                                0AF1  1261 ;       SS$_NOSUCHNODE -  NONEXISTENT NODE OR INVALID CSID
                                0AF1  1262 ;       SS$_ACCVIO  -  ACCESS VIOLATION FOR WRITE DESTINATION
                                0AF1  1263 ;
                                0AF1  1264 ; SIDE EFFECTS:
                                0AF1  1265 ;       NONE
                                0AF1  1266 ;--
                                0AF1  1267
                00000004        0AF1  1268          CSID     = 4                        ; special offset for EXE$NAMCSID
                00000008        0AF1  1269          NODENAME = 8                        ; special offset for EXE$NAMCSID
                                0AF1  1270
                                0AF1  1271 EXE$NAMCSID:                                ; TRANSLATE PNAME TO CSID
                                0AF1  1272          .ENABLE LOCAL_BLOCK
          50    12    DB      0AF1  1273          MFPR    S^#PR$_IPL,R0              ; CHECK THE CURRENT IPL LEVEL
          50    02    D1      0AF4  1274          CMPL    #IPL$_ASTDEL,R0            ; ARE WE ABOVE PAGE FAULT IPL?
                03    1E      0AF7  1275          BGEQU   8$                         ; GOOD, WE CAN FAULT
              00B5    30      0AF9  1276          BSBW    999$                       ; CANNOT BE CALLED ABOVE ASTDEL
54  00000000'EF    D0      0AFC  1277 8$:       MOVL    CLU$GL_CLUB,R4             ; GET THE CLUSTER BLOCK ADDRESS
                03    12      0B03  1278          BNEQU   10$                        ; GOOD, WE'RE IN A CLUSTER
              00A3    31      0B05  1279          BRW     NONEX                      ; CANNOT BE CALLED IF NOT IN A CLUSTER
      54    10 A4     D0      0B08  1280 10$:      MOVL    CLUB$L_LOCAL_CSB(R4),R4    ; GET THE CSB ADDRESS
      50    04 AC     D0      0B0C  1281          MOVL    CSID(AP),R0                ; GET CSID ADDRESS
                1A    13      0B10  1282          BEQL    30$                        ; NO CSID ADDRESS
                              0B12  1283          IFWRT   #4,(R0),20$                ; ERROR IF ACCESS VIOLATION
```

```
                50    0C    3C   0B18  1284 35$:    MOVZWL  #SS$_ACCVIO,R0          ; SET ACCESS VIOLATION ERROR CODE
                                05   0B1B  1285          RSB                     ; AND EXIT
                51    60    D0   0B1C  1286 20$:    MOVL    (R0),R1                 ; NOW FETCH CSID
                      0B    13   0B1F  1287          BEQL    30$                     ; BRANCH IF NO CSID FOUND
                      50    D4   0B21  1288          CLRL    R0                      ; CLEAR CSID ADDRESS,
                                0B23  1289                                          ; DON'T NEED TO REWRITE SAME VALUE
                         009F 31   0B23  1290          BRW     GOTCSID                 ; HAVE THE CSID, GO CHECK IT OUT
                                0B26  1291
          50    0154 8F    3C   0B26  1292 45$:    MOVZWL  #SS$_IVLOGNAM,R0        ; BAD NODENAME STRING
                                05   0B2B  1293          RSB
                                0B2C  1294
                                0B2C  1295 ; NO CSID SPECIFIED (CSIDADR = 0 OR CSIDADR -> 0)
                                0B2C  1296 ;
                                0B2C  1297 ;       R4 -> LOCAL CSB
                                0B2C  1298 ;       R0 = 0 OR R0 -> 0  (CSIDADR = 0 OR CSIDADR -> 0)
                                0B2C  1299 ;       <R1,R2,R3> NOT INTERESTING
                                0B2C  1300
            51    4C A4    D0   0B2C  1301 30$:    MOVL    CSB$L_CSID(R4),R1       ; ASSUME LOCAL CSID
            53    08 AC    D0   0B30  1302          MOVL    NODENAME(AP),R3         ; GET NODENAME ADDRESS IF SPECIFIED
                      03    12   0B34  1303          BNEQ    31$                     ; NEQ MEANS NAME WAS SPECIFIED
                         008C 31   0B36  1304          BRW     GOTCSID                 ; NO NAME SPECIFIED, USE CALLER'S CSID
                                0B39  1305 31$:
                                0B39  1306
                                0B39  1307 ; MUST LOOK UP NODE NAME.  PROBE THE DESCRIPTOR AND THE STRING, AND THEN
                                0B39  1308 ; COPY IT TO THE STACK SO THAT IT CAN BE ACCESSED AFTER WE RAISE IPL.
                                0B39  1309 ;
                                0B39  1310 ;       R4 -> CURRENT CSB
                                0B39  1311 ;       R3 -> NODE NAME DESCRIPTOR (ACCESS NOT YET PROBED)
                                0B39  1312 ;       R0 = 0 OR R0 -> 0  (CSIDADR = 0 OR CSIDADR -> 0)
                                0B39  1313 ;       <R1,R2> NOT INTERESTING
                                0B39  1314
                                0B39  1315          IFNORD  #8,(R3),35$             ; PROBE THE DESCRIPTOR
                52    63    7D   0B3F  1316          MOVQ    (R3),R2                 ; GET THE NODENAME DESCRIPTOR
                52    52    3C   0B42  1317          MOVZWL  R2,R2                   ; IS THE LENGTH LEGAL?
                      DF    13   0B45  1318          BEQL    45$                     ; EQL MEANS NOPE
                52    0F    B1   0B47  1319          CMPW    #15,R2                  ; IS IT TOO LONG?
                      DA    1F   0B4A  1320          BLSSU   45$                     ; LSSU MEANS TOO LONG
                                0B4C  1321          IFNORD  R2,(R3),35$             ; PROBE THE STRING
                5E    10    C2   0B52  1322          SUBL    #16,SP                  ; ALLOCATE BUFFER ON THE STACK
                51    5E    D0   0B55  1323          MOVL    SP,R1                   ; TEMPORARY POINTER TO BUFFER
                      52    DD   0B58  1324          PUSHL   R2                      ; SAVE LENGTH OF NODE NAME STRING
                81    83    90   0B5A  1325 40$:    MOVB    (R3)+,(R1)+             ; COPY NODE NAME STRING FROM USER'S
                   FA 52    F5   0B5D  1326          SOBGTR  R2,40$                  ;     BUFFER ONTO THE STACK
                      52 8ED0   0B60  1327          POPL    R2                      ; RESTORE LENGTH OF NODE NAME STRING
                53    5E    D0   0B63  1328          MOVL    SP,R3                   ; POINTER TO NODE NAME BUFFER
                      50    DD   0B66  1329          PUSHL   R0                      ; SAVE THE CSIDADR ARGUMENT
      50    00000000'EF    3C   0B68  1330          MOVZWL  CLU$GW_MAXINDEX,R0      ; GET THE NUMBER OF ENTRIES
                      50    D7   0B6F  1331          DECL    R0                      ; CONVERT TO HIGHEST OFFSET
                                0B71  1332
                                0B71  1333 ; SCAN CSB VECTOR TO LOOK FOR THIS NODE NAME
                                0B71  1334 ;
                                0B71  1335 ;       R4 -> CURRENT CSB
                                0B71  1336 ;       R3 -> USER'S NODE NAME STRING  (IN BUFFER ON THE STACK)
                                0B71  1337 ;       R2 =  USER'S NODE NAME LENGTH
                                0B71  1338 ;       R0 =  COUNTER FOR CLUSVEC SLOTS
                                0B71  1339
                                0B71  1340 100$:   SETIPL  LOCKPAGE                ; LOCK DOWN THE REST OF THE ROUTINE
```

```
51    00000000'FF40  D0  0B78          MOVL    @CLU$GL_CLUSVEC[R0],R1    ; GET THE POINTER TO THE CSB
                 21  18  0B80          BGEQ    155$                      ; GEQ MEANS UNUSED, TRY THE NEXT ONE
                 51  DD  0B82          PUSHL   R1                        ; SAVE THE POINTER TO THE TARGET CSB
     51    68 A1     D0  0B84          MOVL    CSB$L_SB(R1),R1           ; GET SB ADDRESS
                         0B88
                         0B88   ; IS THIS THE NODENAME?
                         0B88   ;
                         0B88   ;      R0 -   CLUSVEC INDEX
                         0B88   ;      R1 ->  SB (SYSTEM BLOCK)
                         0B88   ;      R2 ->  USER'S NODENAME LENGTH
                         0B88   ;      R3 ->  USER'S NODENAME STRING
                         0B88
                 55  DD  0B88          PUSHL   R5                        ; SAVE R5
     55    44 A1     9E  0B8A          MOVAB   SB$T_NODENAME(R1),R5      ; GET ADDRESS OF NODENAME
          85     52  91  0B8E          CMPB    R2,(R5)+                  ; IS IT THE RIGHT LENGTH?
                 0A  12  0B91          BNEQ    150$                      ; NEQ MEANS NO, TRY THE NEXT ONE
                 0F  BB  0B93          PUSHR   #^M<R0,R1,R2,R3>          ; SAVE REGISTERS FOR THE CMPC3
     65    63    52  29  0B95          CMPC3   R2,(R3),(R5)              ; IS IT THE SAME NODENAME?
                 0F  BA  0B99          POPR    #^M<R0,R1,R2,R3>          ; RESTORE REGISTERS
                 18  13  0B9B          BEQL    GOTNAM                    ; EQL MEANS THIS IS THE ONE
                         0B9D
                         0B9D   ; DID NOT FIND THE NODE BY NAME
                         0B9D
              55 8ED0    0B9D   150$:  POPL    R5                        ; RESTORE R5
              51 8ED0    0BA0          POPL    R1                        ; RESTORE TARGET CSB ADDRESS
          CB 50  F4      0BA3   155$:  SOBGEQ  R0,100$                   ; LOOP IF NOT DONE
             8E  D5      0BA6          TSTL    (SP)+                     ; THROW AWAY R0 FROM STACK
     5E      10  C0      0BA8          ADDL    #16,SP                    ; POP NODE NAME BUFFER FROM STACK
                         0BAB
                         0BAB   ; EXIT WITH NONEXISTENT NODE STATUS
                         0BAB
50      028C 8F  3C      0BAB   NONEX: MOVZWL  #SS$_NOSUCHNODE,R0        ; SET ERROR STATUS
                 05      0BB0          RSB                               ; AND RETURN TO CALLER
                         0BB1
                         0BB1   ; EXIT WITH A CRASH DUMP
                         0BB1
                         0BB1   999$:  BUG_CHECK  ICONCLUDAT,FATAL
                         0BB5
                         0BB5   ; FOUND THE NODE NAME, GET CSID FROM CSB AND CLEAN OFF THE STACK
                         0BB5   ;
                         0BB5   ;      R4 -> CURRENT CSB
                         0BB5
              55 8ED0    0BB5   GOTNAM: POPL   R5                        ; RESTORE R5
              51 8ED0    0BB8          POPL    R1                        ; RESTORE TARGET CSB ADDRESS
     51    4C A1  D0     0BBB          MOVL    CSB$L_CSID(R1),R1         ; GET FULL CSID FOR NAME
              50 8ED0    0BBF          POPL    R0                        ; RESTORE CSIDADR ARGUMENT
     5E      10  C0      0BC2          ADDL    #16,SP                    ; POP NODE NAME BUFFER FROM STACK
                         0BC5
                         0BC5   ; FOUND THE TARGET CSID, VERIFY IT
                         0BC5   ;
                         0BC5   ;      R4 -> CURRENT CSB
                         0BC5   ;      R1 -> CSID OF TARGET NODE
                         0BC5   ;      R0 = 0 OR R0 -> 0  (CSIDADR = 0 OR CSIDADR -> 0)
                         0BC5   ;      <R2,R3> NOT INTERESTING
                         0BC5
                         0BC5   GOTCSID:
                         0BC5          SETIPL  LOCKPAGE                  ; BLOCK SYSTEM EVENTS
```

```
                52    51    3C   0BCC  1398          MOVZWL  R1,R2                          ; EXTRACT NODE INDEX
     00000000'EF      52    B1   0BCF  1399          CMPW    R2,CLU$GW_MAXINDEX           ; TEST AGAINST MAXIMUM VALUE
                      D3    1E   0BD6  1400          BGEQU   NONEX                        ; NONEXISTENT IF GEQU THAN MAXINDEX
52   00000000'FF      42    D0   0BD8  1401          MOVL    @CLU$GL_CLUSVEC[R2],R2       ; GET CSB ADDRESS
                      C9    18   0BE0  1402          BGEQ    NONEX                        ; GEQ MEANS IT'S UNUSED
           4C A2      51    D1   0BE2  1403          CMPL    R1,CSB$L_CSID(R2)            ; CHECK FOR VALID CSID
                      C3    12   0BE6  1404          BNEQ    NONEX                        ; NOT THE SAME
                                 0BE8  1405
                                 0BE8  1406 RETURN:                                     ; SUCCESSFUL EXIT
                54    52    D0   0BE8  1407          MOVL    R2,R4                        ; MOVE CSB ADDRESS OF TARGET
                                 0BEB  1408                                             ; NORMAL STATUS EXIT
                      50    D5   0BEB  1409          TSTL    R0                           ; WAS CSID ADDRESS SPECIFIED
                      0A    13   0BED  1410          BEQL    910$                         ; NO, SKIP STORE OF CSID
                                 0BEF  1411          SETIPL  #IPL$_ASTDEL                 ; ALLOW PAGE FAULTS
                60    51    D0   0BF2  1412          MOVL    R1,(R0)                      ; STORE CSID IN DESTINATION
                      50    D4   0BF5  1413          CLRL    R0                           ; DO NOT WRITE CSID A SECOND TIME
                      CC    11   0BF7  1414          BRB     GOTCSID                      ; MAKE SURE THAT CSID IS STILL VALID
                                 0BF9  1415
                50    01    3C   0BF9  1416 910$:     MOVZWL  #SS$_NORMAL,R0               ; SET SUCCESS STATUS
                      05    0BFC  1417          RSB                                  ; AND RETURN TO CALLER
                                 0BFD  1418
                                 0BFD  1419 ; LOCK THIS PAGE DOWN WHEN WE RAISE IPL
                                 0BFD  1420 ;
                                 0BFD  1421
                                 0BFD  1422 LOCKPAGE:
                      08         0BFD  1423          .BYTE   IPL$_SCS                     ; END OF LOCKED CODE REGION
                                 0BFE  1424          ASSUME  <.-100$> LE 512
                                 0BFE  1425
                                 0BFE  1426          .DISABLE LOCAL_BLOCK
                                 0BFE  1427
                                 0BFE  1428          .END
```

```
$$T1                            = 00000000
ACP$GB_BASEPRIO                   ********    X   02
ACP$GB_DATACHK                    ********    X   02
ACP$GB_MAXREAD                    ********    X   02
ACP$GB_SWAPFLGS                   ********    X   02
ACP$GB_WINDOW                     ********    X   02
ACP$GB_WRITBACK                   ********    X   02
ACP$GW_DINDXCACHE                 ********    X   02
ACP$GW_DIRCACHE                   ********    X   02
ACP$GW_EXTCACHE                   ********    X   02
ACP$GW_EXTLIMIT                   ********    X   02
ACP$GW_FIDCACHE                   ********    X   02
ACP$GW_HDRCACHE                   ********    X   02
ACP$GW_MAPCACHE                   ********    X   02
ACP$GW_QUOCACHE                   ********    X   02
ACP$GW_SYSACC                     ********    X   02
ACP$GW_WORKSET                    ********    X   02
ACP$V_READCHK                   = 00000000  G
ACP$V_SWAPGRP                   = 00000001  G
ACP$V_SWAPMAG                   = 00000003  G
ACP$V_SWAPPRV                   = 00000002  G
ACP$V_SWAPSYS                   = 00000000  G
ACP$V_WRITECHK                  = 00000001  G
ARC$V_CHAR_EMUL                 = 00000004
ARC$V_DCML_EMUL                 = 00000005
ARC$V_DFLT_EMUL                 = 00000008
ARC$V_FFLT_EMUL                 = 00000009
ARC$V_GFLT_EMUL                 = 0000000A
ARC$V_HFLT_EMUL                 = 0000000B
ASTADR                          = 00000018
ASTPRM                          = 0000001C
BIT...                          = 00000001
BITPOS                          = FFFFFFE4
BITSIZ                          = FFFFFFE0
BSTRING                         = 00000001
BUG$_ICONCLUDAT                   ********    X   02
CHECKITEM                         000007BD R      02
CHECK_SPC                         00008A0 R       02
CLU$GB_QDISK                      ********    X   02
CLU$GB_VAXCLUSTER                 ********    X   02
CLU$GL_ALLOCLS                    ********    X   02
CLU$GL_CLUB                       ********    X   02
CLU$GL_CLUSVEC                    ********    X   02
CLU$GW_LCKDIRWT                   ********    X   02
CLU$GW_MAXINDEX                   ********    X   02
CLU$GW_QDSKINTERVAL               ********    X   02
CLU$GW_QDSKVOTES                  ********    X   02
CLU$GW_QUORUM                     ********    X   02
CLU$GW_RECNXINT                   ********    X   02
CLU$GW_VOTES                      ********    X   02
CLUB$B_FSYSID                   = 00000026
CLUB$L_FLAGS                    = 0000001C
CLUB$L_LOCAL_CSB                = 00000010
CLUB$Q_FTIME                    = 0000002C
CLUB$V_CLUSTER                  = 00000000
CLUB$W_NODES                    = 00000024
CLUB$W_QUORUM                   = 00000020
CLUB$W_VOTES                    = 00000022
COMPAT                            000007B1 R      02
CSB$L_CSID                      = 0000004C
CSB$L_SB                        = 00000068
CSB$W_QUORUM                    = 00000052
CSB$W_VOTES                     = 0000
CSID                            = 0000
CSIDADR                         = 00000008
CSTRING                         = 00000002
CTL$GL_PCB                        ********    X   02
EFN                             = 00000004
EXE$                              000007FD R      02
EXE$GETSYI                        00000000 RG     03
EXE$GL_ARCHFLAG                   ********    X   02
EXE$GL_CLITABL                    ********    X   02
EXE$GL_DEFFLAGS                   ********    X   02
EXE$GL_DYNAMIC_FLAGS              ********    X   02
EXE$GL_LOCKRTR?                   ********    X   02
EXE$GL_MSGFLAGS                   ********    X   02
EXE$GL_RTIMESPT                   ********    X   02
EXE$GL_STATIC_FLAGS               ********    X   02
EXE$GL_SYSUIC                     ********    X   02
EXE$GL_WSFLAGS                    ********    X   02
EXE$GQ_BOOTTIME                   ********    X   02
EXE$NAMCSID                       00000AF1 R      02
EXE$PROBEW                        ********    X   02
EXE$V_BRK_DISUSER               = 00000003
EXE$V_BRK_TERM                  = 00000002
EXE$V_BUGDUMP                     ********    X   02
EXE$V_BUGREBOOT                   ********    X   02
EXE$V_CJFLOAD                     ********    X   02
EXE$V_CJFSYSRUJ                   ********    X   02
EXE$V_CLASS_PROT                = 00000000
EXE$V_CONCEALED                   ********    X   02
EXE$V_CRDENABL                    ********    X   02
EXE$V_DISMOUMSG                 = 00000001  G
EXE$V_FATAL_BUG                   ********    X   02
EXE$V_MOUNTMSG                  = 00000000  G
EXE$V_MULTACP                     ********    X   02
EXE$V_NOAUTOCNF                   ********    X   02
EXE$V_NOCLOCK                     ********    X   02
EXE$V_NOCLUSTER                   ********    X   02
EXE$V_OPA0                      = 00000000  G
EXE$V_POOLPGING                   ********    X   02
EXE$V_REBLDSYSD                 = 00000001
EXE$V_RESALLOC                    ********    X   02
EXE$V_SAVEDUMP                    ********    X   02
EXE$V_SBIERR                      ********    X   02
EXE$V_SETTIME                     ********    X   02
EXE$V_SHRF11ACP                   ********    X   02
EXE$V_SSINHIBIT                   ********    X   02
EXE$V_SYSPAGING                   ********    X   02
EXE$V_SYSUAFALT                   ********    X   02
EXE$V_SYSWRTABL                   ********    X   02
EXE$V_WRITESYSPARAMS            = 00000001
EXE$V_XQP_RESIDENT              = 00000000
EXETBL                            00000004 R      02
```

```
EXE_GETSYI                      000006D7 R      02
FETCH_CLU                       00000957 R      02
FLAGS                         = FFFFFFFC
FLD$                            00000808 R      02
FLDTBL                          0000050E R      02
GETSYISW                      = 00000000
GET_SB_FLD                      00000950 R      02
GOTCSID                         00000BC5 R      02
GOTNAM                          00000BB5 R      02
GRET                            0000076A R      02
IOC$GW_LAMAPREG                 ********   X    02
IOC$GW_MAXBUF                   ********   X    02
IOC$GW_MBXBFQUO                 ********   X    02
IOC$GW_MBXMXMSG                 ********   X    02
IOC$GW_MBXNMMSG                 ********   X    02
IOC$GW_MVTIMEOUT                ********   X    02
IOC$GW_XFMXRATE                 ********   X    02
IOSB                          = 00000014
IPL$_ASTDEL                   = 00000002
IPL$_SCS                      = 00000008
ITMLST                        = 00000010
LCK$GL_EXTRASTK                 ********   X    02
LCK$GL_HTBLSIZ                  ********   X    02
LCK$GL_IDTBLMAX                 ********   X    02
LCK$GL_IDTBLSIZ                 ********   X    02
LCK$GL_WAITTIME                 ********   X    02
LNM$GL_HTBLSIZP                 ********   X    02
LNM$GL_HTBLSIZS                 ********   X    02
LOCAL_SB                        0000092A R      02
LOCAL_SPACE                   = FFFFFFE0
LOCK                            0000098D R      02
LOCKPAGE                        00000BFD R      02
MAXCOUNT                        00000000 R      02
MAXSTRUC                      = 00000002
MAX_EXE_ITEM                  = 00000101
MAX_FLD_ITEM                  = 0000002A
MMG$GL_MAXPFIDX                 ********   X    02
MMG$GL_NULLPFL                  ********   X    02
MMG$GL_PAGSWPVC                 ********   X    02
MMG$GL_PHYPGCNT                 ********   X    02
MPW$GB_PRIO                     ********   X    02
MPW$GL_THRESH                   ********   X    02
MPW$GL_WAITLIM                  ********   X    02
MPW$GW_HILIM                    ********   X    02
MPW$GW_LOLIM                    ********   X    02
MPW$GW_MPWPFC                   ********   X    02
NAMCSID                         000009EE R      02
NODE                          = 0000000C
NODENAME                      = 00000008
NONEX                           00000BAB R      02
NULLARG1                      = 00000008
NULLARG2                      = 0000000C
OUTLEN                        = 00000004
PCB$L_PID                     = 00000060
PCB$W_ASTCNT                  = 00000038
PFL$B_FLAGS                   = 00000023
PFL$L_BITMAPSIZ               = 00000014
PFL$L_FREPAGCNT               = 00000018
PFL$V_INITED                  = 00000000
POINT_R4                        00000935 R      02
PQL$GDASTLM                     ********   X    02
PQL$GDBIOLM                     ********   X    02
PQL$GDBYTLM                     ********   X    02
PQL$GDCPULM                     ********   X    02
PQL$GDDIOLM                     ********   X    02
PQL$GDENQLM                     ********   X    02
PQL$GDFILLM                     ********   X    02
PQL$GDJTQUOTA                   ********   X    02
PQL$GDPGFLQUOTA                 ********   X    02
PQL$GDPRCLM                     ********   X    02
PQL$GDTQELM                     ********   X    02
PQL$GDWSDEFAULT                 ********   X    02
PQL$GDWSEXTENT                  ********   X    02
PQL$GDWSQUOTA                   ********   X    02
PQL$GMASTLM                     ********   X    02
PQL$GMBIOLM                     ********   X    02
PQL$GMBYTLM                     ********   X    02
PQL$GMCPULM                     ********   X    02
PQL$GMDIOLM                     ********   X    02
PQL$GMENQLM                     ********   X    02
PQL$GMFILLM                     ********   X    02
PQL$GMJTQUOTA                   ********   X    02
PQL$GMPGFLQUOTA                 ********   X    02
PQL$GMPRCLM                     ********   X    02
PQL$GMTQELM                     ********   X    02
PQL$GMWSDEFAULT                 ********   X    02
PQL$GMWSEXTENT                  ********   X    02
PQL$GMWSQUOTA                   ********   X    02
PR$S_SID_TYPE                 = 00000008
PR$V_SID_TYPE                 = 00000018
PR$_IPL                       = 00000012
PR$_SID                       = 0000003E
PSL$S_PRVMOD                  = 00000002
PSL$V_PRVMOD                  = 00000016
PUTDATA                         00000846 R      02
RETURN                          00000BE8 R      02
SB$B_HWVERS                   = 00000038
SB$B_SYSTEMID                 = 00000018
SB$Q_SWINCARN                 = 0000002C
SB$T_HWTYPE                   = 00000034
SB$T_NODENAME                 = 00000044
SB$T_SWTYPE                   = 00000024
SB$T_SWVERS                   = 00000028
SCH$CLREF                       ********   X    02
SCH$GL_AWSTIME                  ********   X    02
SCH$GL_BORROWLIM                ********   X    02
SCH$GL_CURPCB                   ********   X    02
SCH$GL_GROWLIM                  ********   X    02
SCH$GL_PFRATH                   ********   X    02
SCH$GL_PFRATL                   ********   X    02
SCH$GL_PFRATS                   ********   X    02
SCH$GL_SWPRATE                  ********   X    02
SCH$GL_WSDEC                    ********   X    02
SCH$GL_WSINC                    ********   X    02
```

```
SCH$GW_AWSMIN           ********  X  02
SCH$GW_DORMANTWAIT      ********  X  02
SCH$GW_IOTA             ********  X  02
SCH$GW_LONGWAIT         ********  X  02
SCH$GW_QUAN             ********  X  02
SCH$GW_SWPFAIL          ********  X  02
SCH$POSTEF              ********  X  02
SCS$GA_EXISTS           ********  X  02
SCS$GA_LOCALSB          ********  X  02
SCS$GB_NODENAME         ********  X  02
SCS$GB_PAMXPORT         ********  X  02
SCS$GB_PANOPOLL         ********  X  02
SCS$GB_PANPOLL          ********  X  02
SCS$GB_PASANITY         ********  X  02
SCS$GB_SYSTEMID         ********  X  02
SCS$GB_SYSTEMIDH        ********  X  02
SCS$GB_UDABURST         ********  X  02
SCS$GW_BDTCNT           ********  X  02
SCS$GW_CDTCNT           ********  X  02
SCS$GW_FLOWCUSH         ********  X  02
SCS$GW_MAXDG            ********  X  02
SCS$GW_MAXMSG           ********  X  02
SCS$GW_PAPOLINT         ********  X  02
SCS$GW_PAPOOLIN         ********  X  02
SCS$GW_PAPPDDG          ********  X  02
SCS$GW_PASTMOUT         ********  X  02
SCS$GW_PRCPOLINT        ********  X  02
SCS$GW_RDTCNT           ********  X  02
SGN$GB_KFILSTCT         ********  X  02
SGN$GB_PGTBPFC          ********  X  02
SGN$GB_STARTUP_P1       ********  X  02
SGN$GB_STARTUP_P2       ********  X  02
SGN$GB_STARTUP_P3       ********  X  02
SGN$GB_STARTUP_P4       ********  X  02
SGN$GB_STARTUP_P5       ********  X  02
SGN$GB_STARTUP_P6       ********  X  02
SGN$GB_STARTUP_P7       ********  X  02
SGN$GB_STARTUP_P8       ********  X  02
SGN$GB_SYSPFC           ********  X  02
SGN$GB_TAILORED         ********  X  02
SGN$GL_BALSETCT         ********  X  02
SGN$GL_EXTRACPU         ********  X  02
SGN$GL_EXUSRSTK         ********  X  02
SGN$GL_FREEGOAL         ********  X  02
SGN$GL_FREELIM          ********  X  02
SGN$GL_GBLPAGFIL        ********  X  02
SGN$GL_IRPCNT           ********  X  02
SGN$GL_IRPCNTV          ********  X  02
SGN$GL_LOADFLAGS        ********  X  02
SGN$GL_LRPCNT           ********  X  02
SGN$GL_LRPCNTV          ********  X  02
SGN$GL_LRPMIN           ********  X  02
SGN$GL_LRPSIZE          ********  X  02
SGN$GL_MAXGPGCT         ********  X  02
SGN$GL_MAXVPGCT         ********  X  02
SGN$GL_MAXWSCNT         ********  X  02
SGN$GL_NPAGEDYN         ********  X  02
SGN$GL_NPAGEVIR         ********  X  02
SGN$GL_PAGEDYN          ********  X  02
SGN$GL_PE1              ********  X  02
SGN$GL_PE2              ********  X  02
SGN$GL_PE3              ********  X  02
SGN$GL_PE4              ********  X  02
SGN$GL_PE5              ********  X  02
SGN$GL_PE6              ********  X  02
SGN$GL_SPTREQ           ********  X  02
SGN$GL_SRPCNT           ********  X  02
SGN$GL_SRPCNTV          ********  X  02
SGN$GL_SRPMIN           ********  X  02
SGN$GL_SRPSIZE          ********  X  02
SGN$GL_USER3            ********  X  02
SGN$GL_USER4            ********  X  02
SGN$GL_USERD1           ********  X  02
SGN$GL_USERD2           ********  X  02
SGN$GL_VMS5             ********  X  02
SGN$GL_VMS6             ********  X  02
SGN$GL_VMS7             ********  X  02
SGN$GL_VMS8             ********  X  02
SGN$GL_VMSD1            ********  X  02
SGN$GL_VMSD2            ********  X  02
SGN$GL_VMSD3            ********  X  02
SGN$GL_VMSD4            ********  X  02
SGN$GW_CTLIMGLIM        ********  X  02
SGN$GW_CTLPAGES         ********  X  02
SGN$GW_DFPFC            ********  X  02
SGN$GW_GBLSECNT         ********  X  02
SGN$GW_IMGIOCNT         ********  X  02
SGN$GW_ISPPGCT          ********  X  02
SGN$GW_MAXPRCCT         ********  X  02
SGN$GW_MAXPSTCT         ********  X  02
SGN$GW_MINWSCNT         ********  X  02
SGN$GW_PAGFILCT         ********  X  02
SGN$GW_PCHANCNT         ********  X  02
SGN$GW_PIOPAGES         ********  X  02
SGN$GW_PIXSCAN          ********  X  02
SGN$GW_SWPFILCT         ********  X  02
SGN$GW_SWPFILES         ********  X  02
SGN$GW_SYSDWSCT         ********  X  02
SGN$GW_TPWAIT           ********  X  02
SGN$GW_WSLMXSKP         ********  X  02
SGN$V_LOADCHKPRT      = 00000001  G
SGN$V_LOADERAPAT      = 00000000  G
SGN$V_LOADMTACCESS    = 00000002  G
SIZ...                = 00000001
SPC_CLUB                00000901  R  02
SPC_CSB                 00000913  R  02
SPC_EXISTS              000008D4  R  02
SPC_LOCK              = 00000918  R  02
SPC_MEMBER              000008F9  R  02
SPC_NEGATIVE            000008EF  R  02
SPC_PAGESWAP            0000098E  R  02
SPC_PROCREG             000008E5  R  02
SPC_SB                  00000939  R  02
SPECIAL                 0000063B  R  02
```

```
SPECIAL_LEN                     = 0000001A
SPECIAL_SPACE                   = FFFFFFEC
SS$_ACCVIO                      = 0000000C
SS$_BADPARAM                    = 00000014
SS$_EXASTLM                     = 00002A04
SS$_IVLOGNAM                    = 00000154
SS$_NOMORENODE                  = 00000A00
SS$_NORMAL                      = 00000001
SS$_NOSUCHNODE                  = 0000028C
STEP                            = 00000005
SWP$GB_PRIO                       ********    X   02
SWP$GL_SWPPGCNT                   ********    X   02
SWP$GW_SWPINC                     ********    X   02
SYI$C_FLDTYPE                   = 00000002
SYI$_ACP_BASEPRIO               = 000010B5
SYI$_ACP_DATACHECK              = 000010B4
SYI$_ACP_DINDXCACHE             = 00001101
SYI$_ACP_DIRCACHE               = 000010AA
SYI$_ACP_EXTCACHE               = 000010AD
SYI$_ACP_EXTLIMIT               = 000010AE
SYI$_ACP_FIDCACHE               = 000010AC
SYI$_ACP_HDRCACHE               = 000010A9
SYI$_ACP_MAPCACHE               = 000010A8
SYI$_ACP_MAXREAD                = 000010B1
SYI$_ACP_MULTIPLE               = 00002005
SYI$_ACP_QUOCACHE               = 000010AF
SYI$_ACP_REBLDSYSD              = 00002029
SYI$_ACP_SHARE                  = 0000200C
SYI$_ACP_SWAPFLGS               = 000010B6
SYI$_ACP_SYSACC                 = 000010B0
SYI$_ACP_WINDOW                 = 000010B2
SYI$_ACP_WORKSET                = 000010AB
SYI$_ACP_WRITEBACK              = 000010B3
SYI$_ACP_XQP_RES                = 00002025
SYI$_ALLOCLASS                  = 000010E5
SYI$_ARCHFLAG                   = 000010DA
SYI$_AWSMIN                     = 00001038
SYI$_AWSTIME                    = 00001039
SYI$_BALSETCNT                  = 00001012
SYI$_BJOBLIM                    = 000010B9
SYI$_BOOTTIME                   = 000010BF
SYI$_BORROWLIM                  = 00001056
SYI$_BUGCHECKFATAL              = 00002004
SYI$_BUGREBOOT                  = 00002001
SYI$_CHANNELCNT                 = 00001024
SYI$_CHARACTER_EMULATED         = 0000201E
SYI$_CJFLOAD                    = 00002019
SYI$_CJFSYSRUJ                  = 0000201A
SYI$_CLASS_PROT                 = 0000201D
SYI$_CLISYMTBL                  = 0000105B
SYI$_CLUSTER_FSYSID             = 000010CD
SYI$_CLUSTER_FTIME              = 000010CE
SYI$_CLUSTER_MEMBER             = 000010CF
SYI$_CLUSTER_NODES              = 000010CA
SYI$_CLUSTER_QUORUM             = 000010CC
SYI$_CLUSTER_VOTES              = 000010CB
SYI$_CONCEAL_DEVICES            = 00002012
SYI$_CPU                        = 00002000
SYI$_CRDENABLE                  = 00002002
SYI$_CTLIMGLIM                  = 00001027
SYI$_CTLPAGES                   = 00001026
SYI$_DEADLOCK_WAIT              = 0000105E
SYI$_DECIMAL_EMULATED           = 0000201F
SYI$_DEFMBXBUFQUO               = 00001050
SYI$_DEFMBXMXMSG                = 00001051
SYI$_DEFMBXNUMMSG               = 00001052
SYI$_DEFPRI                     = 000010B7
SYI$_DEFQUEPRI                  = 000010E2
SYI$_DISK_QUORUM                = 000010DC
SYI$_DISMOUMSG                  = 00002015
SYI$_DLCKEXTRASTK               = 00001011
SYI$_DORMANTWAIT                = 000010F1
SYI$_DUMPBUG                    = 00002003
SYI$_D_FLOAT_EMULATED           = 00002020
SYI$_EXTRACPU                   = 0000104C
SYI$_EXUSRSTK                   = 0000101B
SYI$_FREEGOAL                   = 00001054
SYI$_FREELIM                    = 00001053
SYI$_F_FLOAT_EMULATED           = 00002021
SYI$_GBLPAGES                   = 00001007
SYI$_GBLPAGFIL                  = 00001008
SYI$_GBLSECTIONS                = 00001006
SYI$_GROWLIM                    = 00001055
SYI$_G_FLOAT_EMULATED           = 00002022
SYI$_H_FLOAT_EMULATED           = 00002023
SYI$_IJOBLIM                    = 000010B8
SYI$_IMGIOCNT                   = 00001028
SYI$_INTSTKPAGES                = 00001010
SYI$_IOTA                       = 0000103D
SYI$_IRPCOUNT                   = 00001013
SYI$_IRPCOUNTV                  = 00001014
SYI$_KFILSTCNT                  = 00001005
SYI$_LAMAPREGS                  = 00001059
SYI$_LASTEXE                    = 00001102
SYI$_LASTFLD                    = 0000202B
SYI$_LGI_BRK_DISUSER            = 00002028
SYI$_LGI_BRK_LIM                = 000010E8
SYI$_LGI_BRK_TERM               = 00002027
SYI$_LGI_BRK_TMO                = 000010E9
SYI$_LGI_HID_TIM                = 000010EA
SYI$_LGI_PWD_TMO                = 000010EB
SYI$_LGI_RETRY_LIM              = 000010E6
SYI$_LGI_RETRY_TMO              = 000010E7
SYI$_LNMPHASHTBL                = 00001072
SYI$_LNMSHASHTBL                = 00001071
SYI$_LOADCHKPRT                 = 00002017
SYI$_LOADERAPT                  = 00002016
SYI$_LOADMTACCESS               = 00002024
SYI$_LOCKDIRWT                  = 000010EF
SYI$_LOCKIDTBL                  = 0000105C
SYI$_LOCKIDTBL_MAX              = 000010C0
SYI$_LOCKRETRY                  = 00001057
SYI$_LONGWAIT                   = 0000103E
SYI$_LRPCOUNT                   = 0000101C
```

```
SYI$_LRPCOUNTV                  =  0000101D
SYI$_LRPMIN                     =  0000101F
SYI$_LRPSIZE                    =  0000101E
SYI$_MAXBUF                     =  0000104F
SYI$_MAXPROCESSCNT              =  00001009
SYI$_MAXQUEPRI                  =  000010E3
SYI$_MAXSYSGROUP                =  0000104D
SYI$_MINWSCNT                   =  0000100C
SYI$_MOUNTMSG                   =  00002014
SYI$_MPW_HILIMIT                =  0000102B
SYI$_MPW_LOLIMIT                =  0000102C
SYI$_MPW_PRIO                   =  0000102D
SYI$_MPW_THRESH                 =  0000102F
SYI$_MPW_WAITLIMIT              =  00001030
SYI$_MPW_WRTCLUSTER             =  0000102A
SYI$_MVTIMEOUT                  =  0000104E
SYI$_NJOBLIM                    =  000010BA
SYI$_NOAUTOCONFIG               =  00002006
SYI$_NOCLOCK                    =  00002007
SYI$_NOCLUSTER                  =  00002008
SYI$_NODENAME                   =  000010D9
SYI$_NODE_AREA                  =  0000201B
SYI$_NODE_CSID                  =  000010D0
SYI$_NODE_HWTYPE                =  000010D7
SYI$_NODE_HWVERS                =  000010D8
SYI$_NODE_NUMBER                =  0000201C
SYI$_NODE_QUORUM                =  000010D2
SYI$_NODE_SWINCARN              =  000010D4
SYI$_NODE_SWTYPE                =  000010D5
SYI$_NODE_SWVERS                =  000010D6
SYI$_NODE_SYSTEMID              =  000010D3
SYI$_NODE_VOTES                 =  000010D1
SYI$_NPAGEDYN                   =  00001016
SYI$_NPAGEVIR                   =  00001017
SYI$_OLDCPU                     =  00000200
SYI$_OLDSID                     =  00000201
SYI$_OLDVERSION                 =  00000100
SYI$_PAGEDYN                    =  00001018
SYI$_PAGEFILE_FREE              =  000010F4
SYI$_PAGEFILE_PAGE              =  000010F2
SYI$_PAGFILCNT                  =  0000100D
SYI$_PAGTBLPFC                  =  00001003
SYI$_PAMAXPORT                  =  000010E0
SYI$_PANOPOLL                   =  000010F8
SYI$_PAHUMPOLL                  =  0000106C
SYI$_PAPOLLINTERVAL             =  0000106D
SYI$_PAPOOLINTERVAL             =  0000106E
SYI$_PASANITY                   =  000010E1
SYI$_PASTDGBUF                  =  0000106B
SYI$_PASTIMOUT                  =  0000106A
SYI$_PE1                        =  000010F9
SYI$_PE2                        =  000010FA
SYI$_PE3                        =  000010FB
SYI$_PE4                        =  000010FC
SYI$_PE5                        =  000010FD
SYI$_PE6                        =  000010FE
SYI$_PFCDEFAULT                 =  00001002
SYI$_PFRATH                     =  00001034
SYI$_PFRATL                     =  00001033
SYI$_PFRATS                     =  00001035
SYI$_PHYSICALPAGES              =  00001032
SYI$_PIOPAGES                   =  00001025
SYI$_PIXSCAN                    =  0000100A
SYI$_POOLPAGING                 =  00002009
SYI$_PQL_DASTLM                 =  0000108E
SYI$_PQL_DBIOLM                 =  00001090
SYI$_PQL_DBYTLM                 =  00001092
SYI$_PQL_DCPULM                 =  00001094
SYI$_PQL_DDIOLM                 =  00001096
SYI$_PQL_DENQLM                 =  000010A6
SYI$_PQL_DFILLM                 =  00001098
SYI$_PQL_DJTQUOTA               =  000010EC
SYI$_PQL_DPGFLQUOTA             =  0000109A
SYI$_PQL_DPRCLM                 =  0000109C
SYI$_PQL_DTQELM                 =  0000109E
SYI$_PQL_DWSDEFAULT             =  000010A0
SYI$_PQL_DWSEXTENT              =  000010A4
SYI$_PQL_DWSQUOTA               =  000010A2
SYI$_PQL_MASTLM                 =  0000108F
SYI$_PQL_MBIOLM                 =  00001091
SYI$_PQL_MBYTLM                 =  00001093
SYI$_PQL_MCPULM                 =  00001095
SYI$_PQL_MDIOLM                 =  00001097
SYI$_PQL_MENQLM                 =  000010A7
SYI$_PQL_MFILLM                 =  00001099
SYI$_PQL_MJTQUOTA               =  000010ED
SYI$_PQL_MPGFLQUOTA             =  0000109B
SYI$_PQL_MPRCLM                 =  0000109D
SYI$_PQL_MTQELM                 =  0000109F
SYI$_PQL_MWSDEFAULT             =  000010A1
SYI$_PQL_MWSEXTENT              =  000010A5
SYI$_PQL_MWSQUOTA               =  000010A3
SYI$_PRCPOLINTERVAL             =  00001069
SYI$_PROCSECTCNT                =  0000100B
SYI$_QDSKINTERVAL               =  000010E4
SYI$_QDSKVOTES                  =  000010F0
SYI$_QUANTUM                    =  00001029
SYI$_QUORUM                     =  000010BC
SYI$_REALTIME_SPTS              =  0000105A
SYI$_RECNXINTERVAL              =  000010BE
SYI$_RESALLOC                   =  00002010
SYI$_RESHASHTBL                 =  0000105D
SYI$_RJOBLIM                    =  000010BB
SYI$_RMS_DFMBC                  =  00001084
SYI$_RMS_DFMBFHSH               =  0000108A
SYI$_RMS_DFMBFIDX               =  00001089
SYI$_RMS_DFMBFREL               =  00001088
SYI$_RMS_DFMBFSDK               =  00001085
SYI$_RMS_DFMBFSMT               =  00001086
SYI$_RMS_DFMBFSUR               =  00001087
SYI$_RMS_DFNBC                  =  00001100
SYI$_RMS_EXTEND_SIZE            =  0000108C
SYI$_RMS_FILEPROT               =  0000108D
SYI$_RMS_GBLBUFQUO              =  000010FF
```

```
SYI$_RMS_PROLOGUE               = 0000108B
SYI$_SAVEDUMP                   = 00002013
SYI$_SBIERRENABLE               = 0000200A
SYI$_SCSBUFFCNT                 = 0000105F
SYI$_SCSCONNCNT                 = 00001060
SYI$_SCSFLOWCUSH                = 00001064
SYI$_SCSMAXDG                   = 00001062
SYI$_SCSMAXMSG                  = 00001063
SYI$_SCSNODE                    = 00001067
SYI$_SCSRESPCNT                 = 00001061
SYI$_SCSSYSTEMID                = 00001065
SYI$_SCSSYSTEMIDH               = 00001066
SYI$_SCS_EXISTS                 = 000010DB
SYI$_SETTIME                    = 0000200B
SYI$_SID                        = 00001001
SYI$_SPTREQ                     = 0000101A
SYI$_SRPCOUNT                   = 00001020
SYI$_SRPCOUNTV                  = 00001021
SYI$_SRPMIN                     = 00001023
SYI$_SRPSIZE                    = 00001022
SYI$_SSINHIBIT                  = 00002011
SYI$_STARTUP_P1                 = 000010C2
SYI$_STARTUP_P2                 = 000010C3
SYI$_STARTUP_P3                 = 000010C4
SYI$_STARTUP_P4                 = 000010C5
SYI$_STARTUP_P5                 = 000010C6
SYI$_STARTUP_P6                 = 000010C7
SYI$_STARTUP_P7                 = 000010C8
SYI$_STARTUP_P8                 = 000010C9
SYI$_SWAPFILE_FREE              = 000010F5
SYI$_SWAPFILE_PAGE              = 000010F3
SYI$_SWPALLOCINC                = 0000103C
SYI$_SWPFAIL                    = 0000103F
SYI$_SWPFILCNT                  = 0000100E
SYI$_SWPOUTPGCNT                = 0000103B
SYI$_SWPRATE                    = 0000103A
SYI$_SWP_PRIO                   = 0000102E
SYI$_SYSMWCNT                   = 0000100F
SYI$_SYSPAGING                  = 0000200D
SYI$_SYSPFC                     = 00001004
SYI$_TAILORED                   = 000010C1
SYI$_TBSKIPWSL                  = 00001031
SYI$_TIMEPROMPTWAIT             = 0000106F
SYI$_TTY_ALTALARM               = 0000107D
SYI$_TTY_ALTYPAHD               = 0000107C
SYI$_TTY_AUTOCHAR               = 000010F7
SYI$_TTY_BUF                    = 00001078
SYI$_TTY_CLASSNAME              = 00001081
SYI$_TTY_DEFCHAR                = 00001079
SYI$_TTY_DEFCHAR2               = 0000107A
SYI$_TTY_DEFPORT                = 00001083
SYI$_TTY_DIALTYPE               = 00001074
SYI$_TTY_DMASIZE                = 0000107E
SYI$_TTY_OWNER                  = 00001080
SYI$_TTY_PARITY                 = 00001077
SYI$_TTY_PROT                   = 0000107F
SYI$_TTY_RSPEED                 = 00001076
SYI$_TTY_SCANDELTA              = 00001073
SYI$_TTY_SILOTIME               = 00001082
SYI$_TTY_SPEED                  = 00001075
SYI$_TTY_TIMEOUT                = 000010F6
SYI$_TTY_TYPAHDSZ               = 0000107B
SYI$_UAFALTERNATE               = 0000200E
SYI$_UDABURSTRATE               = 00001070
SYI$_USER3                      = 0000104A
SYI$_USER4                      = 0000104B
SYI$_USERD1                     = 00001048
SYI$_USERD2                     = 00001049
SYI$_VAXCLUSTER                 = 000010EE
SYI$_VERSION                    = 00001000
SYI$_VIRTUALPAGECNT             = 00001019
SYI$_VMS5                       = 00001044
SYI$_VMS6                       = 00001045
SYI$_VMS7                       = 00001046
SYI$_VMS8                       = 00001047
SYI$_VMSD1                      = 00001040
SYI$_VMSD2                      = 00001041
SYI$_VMSD3                      = 00001042
SYI$_VMSD4                      = 00001043
SYI$_VOTES                      = 000010BD
SYI$_WRITABLESYS                = 0000200F
SYI$_WRITESYSPARAMS             = 00002026
SYI$_WSDEC                      = 00001037
SYI$_WSINC                      = 00001036
SYI$_WSMAX                      = 00001015
SYI$_WS_OPAO                    = 0000202A
SYI$_XFMAXRATE                  = 00001058
SYI_BIT                         = 00000004
SYI_S_INCLUSTER                 = 00000001
SYI_S_REMOTE_NODE               = 00000001
SYI_S_RETIRED                   = 00000001
SYI_S_WILD                      = 00000001
SYI_V_INCLUSTER                 = 00000001
SYI_V_REMOTE_NODE               = 00000002
SYI_V_RETIRED                   = 00000003
SYI_V_WILD                      = 00000000
SYS$DCLAST                      ******** GX    02
SYS$GB_BRK_LIM                  ********  X    02
SYS$GB_DEFPRI                   ********  X    02
SYS$GB_DEFQUEPRI                ********  X    02
SYS$GB_DFMBC                    ********  X    02
SYS$GB_DFMBFHSH                 ********  X    02
SYS$GB_DFMBFIDX                 ********  X    02
SYS$GB_DFMBFREL                 ********  X    02
SYS$GB_DFMBFSDK                 ********  X    02
SYS$GB_DFMBFSMT                 ********  X    02
SYS$GB_DFMBFSUR                 ********  X    02
SYS$GB_DFNBC                    ********  X    02
SYS$GB_MAXQUEPRI                ********  X    02
SYS$GB_PWD_TMO                  ********  X    02
SYS$GB_RETRY_LIM                ********  X    02
SYS$GB_RETRY_TMO                ********  X    02
SYS$GB_RMSPROLOG                ********  X    02
SYS$GL_BRK_TMO                  ********  X    02
```

```
SYS$GL_HID_TIM                  ********    X   02
SYS$GQ_VERSION                  ********    X   02
SYS$GW_BJOBLIM                  ********    X   02
SYS$GW_FILEPROT                 ********    X   02
SYS$GW_GBLBUFQUO                ********    X   02
SYS$GW_IJOBLIM                  ********    X   02
SYS$GW_NJOBLIM                  ********    X   02
SYS$GW_RJOBLIM                  ********    X   02
SYS$GW_RMSEXTEND                ********    X   02
TEMPORARY                     = FFFFFFE8
TTY$GB_AUTOCHAR                 ********    X   02
TTY$GB_DEFSPEED                 ********    X   02
TTY$GB_DIALTYP                  ********    X   02
TTY$GB_PARITY                   ********    X   02
TTY$GB_RSPEED                   ********    X   02
TTY$GB_SILOTIME                 ********    X   02
TTY$GL_DEFCHAR                  ********    X   02
TTY$GL_DEFCHAR2                 ********    X   02
TTY$GL_DEFPORT                  ********    X   02
TTY$GL_DELTA                    ********    X   02
TTY$GL_OWNUIC                   ********    X   02
TTY$GL_TIMEOUT                  ********    X   02
TTY$GW_ALTALARM                 ********    X   02
TTY$GW_ALTYPAHD                 ********    X   02
TTY$GW_CLASSNAM                 ********    X   02
TTY$GW_DEFBUF                   ********    X   02
TTY$GW_DMASIZE                  ********    X   02
TTY$GW_PROT                     ********    X   02
TTY$GW_TYPAHDSZ                 ********    X   02
VALUE                         = 00000000
VERIFY_CSB                      00000964 R      02
XTYPE                         = 00000001
```

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
YF$$SYSGETSYI	00000BFE (3070.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
YEXEPAGED	00000005 (5.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	0000000A (10.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	39	00:00:00.06	00:00:00.59
Command processing	130	00:00:00.70	00:00:04.42
Pass 1	1222	00:01:21.37	00:03:18.24
Symbol table sort	0	00:00:02.71	00:00:06.24
Pass 2	790	00:00:16.21	00:00:37.57
Symbol table output	83	00:00:00.52	00:00:01.89

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	2268	00:01:41.60	00:04:08.98

The working set limit was 3000 pages.
439180 bytes (858 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1682 non-local and 67 local symbols.
1428 source lines were read in Pass 1, producing 54 object records in Pass 2.
136 pages of virtual memory were used to define 37 macros.

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

Macro library name	Macros defined
_255DUA28:[SYSLIB]SYSBLDMLB.MLB;1	5
_255DUA28:[SYS.OBJ]LIB.MLB;1	14
_255DUA28:[SYSLIB]STARLET.MLB;2	12
TOTALS (all libraries)	31

4130 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SYSGETSYI/OBJ=OBJ$:SYSGETSYI MSRC$:SYSGETSYI/UPDATE=(ENH$:SYSGETSYI)+EXECML$/LIB+SYS$LIBRARY:SYSBLDMLB/LIB
```

SYSGETSYI
LIS

SYSGETPTI
LIS

SYSTMGFIX
LIS

SYSGETTIM
LIS

SYSGETLKI
LIS

SYSGETMSG
LIS

SYSIMGACT
LIS