```
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGG   RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGG   RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGG   RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
SSS              MMMMMM   MMMMMM  GGG              RRR         RRR        TTT        LLL
SSS              MMMMMM   MMMMMM  GGG              RRR         RRR        TTT        LLL
SSS              MMMMMM   MMMMMM  GGG              RRR         RRR        TTT        LLL
SSS              MMM   MMM   MMM  GGG              RRR         RRR        TTT        LLL
SSS              MMM   MMM   MMM  GGG              RRR         RRR        TTT        LLL
SSS              MMM   MMM   MMM  GGG              RRR         RRR        TTT        LLL
   SSSSSSSSS     MMM         MMM  GGG              RRRRRRRRRRRR           TTT        LLL
   SSSSSSSSS     MMM         MMM  GGG              RRRRRRRRRRRR           TTT        LLL
   SSSSSSSSS     MMM         MMM  GGG              RRRRRRRRRRRR           TTT        LLL
            SSS  MMM         MMM  GGG   GGGGGGGGG  RRR   RRR              TTT        LLL
            SSS  MMM         MMM  GGG   GGGGGGGGG  RRR   RRR              TTT        LLL
            SSS  MMM         MMM  GGG   GGGGGGGGG  RRR   RRR              TTT        LLL
            SSS  MMM         MMM  GGG         GGG  RRR      RRR           TTT        LLL
            SSS  MMM         MMM  GGG         GGG  RRR      RRR           TTT        LLL
            SSS  MMM         MMM  GGG         GGG  RRR      RRR           TTT        LLL
SSSSSSSSSSSS     MMM         MMM     GGGGGGGGG     RRR         RRR        TTT        LLLLLLLLLLLLLLL
SSSSSSSSSSSS     MMM         MMM     GGGGGGGGG     RRR         RRR        TTT        LLLLLLLLLLLLLLL
SSSSSSSSSSSS     MMM         MMM     GGGGGGGGG     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
_$2
Val
---
001
001
001
001
001
001
001
001
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

FILEID**SMGUSRTRM

C 1

```
  SSSSSSSS  MM      MM    GGGGGGGG  UU      UU    SSSSSSSS  RRRRRRRR    TTTTTTTTTT  RRRRRRRR    MM      MM
  SSSSSSSS  MM      MM    GGGGGGGG  UU      UU    SSSSSSSS  RRRRRRRR    TTTTTTTTTT  RRRRRRRR    MM      MM
SS          MMMM  MMMM  GG          UU      UU  SS          RR      RR      TT      RR      RR  MMMM  MMMM
SS          MMMM  MMMM  GG          UU      UU  SS          RR      RR      TT      RR      RR  MMMM  MMMM
SS          MM  MM  MM  GG          UU      UU  SS          RR      RR      TT      RR      RR  MM  MM  MM
SS          MM  MM  MM  GG          UU      UU  SS          RR      RR      TT      RR      RR  MM  MM  MM
  SSSSSS    MM      MM  GG          UU      UU    SSSSSS    RRRRRRRR        TT      RRRRRRRR    MM      MM
  SSSSSS    MM      MM  GG          UU      UU    SSSSSS    RRRRRRRR        TT      RRRRRRRR    MM      MM
        SS  MM      MM  GG  GGGGGG  UU      UU          SS  RR  RR          TT      RR  RR      MM      MM
        SS  MM      MM  GG  GGGGGG  UU      UU          SS  RR  RR          TT      RR  RR      MM      MM
        SS  MM      MM  GG      GG  UU      UU          SS  RR    RR        TT      RR    RR    MM      MM      ....
        SS  MM      MM  GG      GG  UU      UU          SS  RR    RR        TT      RR    RR    MM      MM      ....
SSSSSSSS    MM      MM    GGGGGG    UUUUUUUUUU  SSSSSSSS    RR      RR      TT      RR      RR  MM      MM      ....
SSSSSSSS    MM      MM    GGGGGG    UUUUUUUUUU  SSSSSSSS    RR      RR      TT      RR      RR  MM      MM      ....

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
```

```
0 %TITLE 'SMG$INTERFACE_TERM_TABLE - User Interface to Termtable'
0 MODULE SMG$INTERFACE_TERM_TABLE (
0                 IDENT = 'T-005' ! File: SMGUSRTRM.B32 Edit: STAN1005
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:      Screen Management
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains routines which return device-specific
1 !       information about terminals, including, but not
1 !       limited to, the proper character sequences to perform
1 !       sundry functions.
1 !
1 ! ENVIRONMENT:  User mode, Shared library routines.
1 !
1 ! AUTHOR: P. Levesque, CREATION DATE: 9-Nov-1983
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-005 - Remove use of FNM= construct which produces code that
1 !         is not appropriate for use within a shared image.
1 !         STAN 18-Mar-1984.
1 ! 1-004 - When fetching the terminal type from a private section,
1 !         'unbias' the number.  PLL 22-Feb-1984
1 ! 1-003 - Allow shared use of private sections. STAN. 14-Feb-1984.
1 ! 1-002 - Allow positive terminal types. STAN. 8-Feb-1984.
1 ! 1-001 - Original.  PLL 9-Nov-1983
1 !--
```

```
%SBTTL 'Declarations'
!
! SWITCHES:
!

!
! LINKAGES:
!
!     NONE
!
! INCLUDE FILES:
!

REQUIRE 'SRC$:SMGPROLOG';                  ! defines psects, macros, etc.

LIBRARY 'RTLML:SMGTPALIB';                 ! TERMTABLE macros, defs, etc

!
! TABLE OF CONTENTS:
!

FORWARD ROUTINE

    SMG$INIT_TERM_TABLE,                   ! Setup term table for future calls
    SMG$INIT_TERM_TABLE_BY_TYPE,           ! Same as above except input is
                                           ! device type instead of device name
    SMG$GET_TERM_DATA,                     ! Get terminal characteristics
    SMG$DEL_TERM_TABLE,                    ! Delete term table

    MAP_PRIV_SECTION,                      ! Map private TERMTABLE.EXE as a
                                           ! temporary section
    MAP_GBL_SECTION,                       ! Map to system TERMTABLE.EXE in
                                           ! global section
    SEARCH_FOR_TERM_DEF,                   ! Search an index for a given terminal
                                           ! name
    SEARCH_FOR_TERM_DEF_BY_TYPE,           ! Search an index for a given terminal
                                           ! type
    DECODE_ARITH_STRING;                   ! Decode arithmetic data

!
! MACROS:
!
!

MACRO $BIAS_TYPE_NUMBER (TERM_TAB, TYPE, LENGTH, BUFFER) =
!+
!       If this definition is from a private table, terminal type
!       numbers must be biased so that they are in the range -65 to -127.
!       (Note that this bias can't be applied at compile time because it's
!       not known whether the table will be mapped privately or globally.)
!
!       Since we can't modify the type number in the definition (the section
!       is read only), we must copy the type number directly into the user's
!       buffer.
!-
BEGIN
LOCAL
```

```
    BIAS_TYPE;

!+
! Compare the terminal definition address with the starting address of
! the private section.  If there is no private section or the terminal
! definition is at a lower address, then this must be a global definition.
!-
    BIAS_TYPE = TYPE;                       ! init to no bias

    IF .PRIV_SECTION_ADDR NEQ 0 AND
       TERM_TAB GEQ .PRIV_SECTION_ADDR
    THEN
        !+
        ! Don't bias the type of a DEC terminal (which will have
        ! a positive number).
        !-
        IF TYPE LSS 0
        THEN
            BIAS_TYPE = .BIAS_TYPE + K_PRIV_TYPE;

    CH$MOVE (LENGTH, BIAS_TYPE, BUFFER);
                                    ! put in the user's buffer
END%;

MACRO $UNBIAS_TYPE_NUMBER (TERM_TAB, TYPE) =
BEGIN
!+
! Compare the terminal definition address with the starting address of
! the private section.  If there is no private section or the terminal
! definition is at a lower address, then this must be a global definition.
!-
    IF .PRIV_SECTION_ADDR NEQ 0 AND
       TERM_TAB GEQ .PRIV_SECTION_ADDR
    THEN
        !+
        ! Don't bias the type of a DEC terminal (which will have
        ! a positive number).
        !-
        IF .TYPE LSS 0
        THEN
            TYPE = .TYPE - K_PRIV_TYPE;

END;%;

!+
! Some string capabilities expect parameters.  If the caller does not
! supply any parameters, we default them to 1's.  The caller must supply
! all or none of the parameters.
!-
MACRO $CHECK_SUFFICIENT_USER_ARGS =
BEGIN
    IF NOT NULLPARAMETER (INPUT_ARG_VECTOR)
    THEN
        IF .INPUT_ARG_VECTOR [0] NEQ 0
        THEN
            BEGIN
            IF .INPUT_ARG_VECTOR [0] NEQ .SMG$$NUM_PARAMS [..REQUEST_CODE]
```

```
                THEN
                    RETURN (SMG$_NO_ARGS);  ! require all or none

                ! Caller provided args - overwrite defaults

                INCR CTR FROM 0 TO .SMG$$NUM_PARAMS [..REQUEST_CODE] - 1 DO
                    LOCAL_ARGS [.CTR] = .INPUT_ARG_VECTOR [.CTR+1];
                END;
END;%;

!
! EQUATED SYMBOLS:
!

!
! FIELDS:
!
!       NONE
!
! PSECTS:
!

!
! STRUCTURES:
!


!
! OWN STORAGE:
!

!+
! The following are used to avoid mapping a section twice.
!-

OWN
    GBL_SECTION_ADDR : INITIAL (0) VOLATILE,
    PRIV_SECTION_ADDR : INITIAL (0) VOLATILE,
    PRIV_SECTION_ADDR_END : INITIAL (0) VOLATILE,
    PRIV_SECTION_CHANNEL : INITIAL (0) VOLATILE;

!+
! We need the byte equivalents of some longword constants.
! Bliss doesn't think it's possible for a byte to be equal
! to a longword and it is much more efficient for us to
! deal with bytes.
!-
OWN
    K_OPERAND : BYTE INITIAL (SMG$K_OPERAND <0,8>),
    K_SUBSTITUTE : BYTE INITIAL (SMG$K_SUBSTITUTE <0,8>),
    K_ADD : BYTE INITIAL (SMG$K_ADD <0,8>),
    K_SUBTRACT : BYTE INITIAL (SMG$K_SUBTRACT <0,8>),
    K_MULTIPLY : BYTE INITIAL (SMG$K_MULTIPLY <0,8>),
    K_DIVIDE : BYTE INITIAL (SMG$K_DIVIDE <0,8>),
    K_STORE : BYTE INITIAL (SMG$K_STORE <0,8>);

!
```

```
! Literals used in various routines
!

LITERAL
    K_MATCH = 0,
    K_OFF = 0,
    K_ON = 1,
    K_PRIV_TYPE = -65;              ! private types are -65 to -127,
                                    ! system types are -2 to -64

!
! EXTERNAL REFERENCES:
!

EXTERNAL ROUTINE
    SMG$$NUMBER_PARAMETERS,                 ! init # params for caps
    STR$COPY_DX,                            ! copy source string to destination
    STR$CASE_BLIND_COMPARE;                 ! compare 2 strings without
                                            ! regard for upper/lower case

EXTERNAL
    SMG$_WRONUMARG,         ! wrong number of arguments
    SMG$_TABID_MIS,         ! Termtable Id Mismatch
    SMG$_UNDTERNAM,         ! Undefined terminal name
    SMG$_UNDTERTYP,         ! Undefined terminal type
    SMG$_PRISECMAP,         ! Private terminal table used
    SMG$_GBLSECMAP,         ! System terminal table used
    SMG$_UNDTERNOP,         ! Undefined terminal - no private termtable
    SMG$_UNDTERNOS,         ! Undefined terminal - no system termtable
    SMG$_INVTERTAB,         ! Invalid Termtable
    SMG$_INVREQCOD,         ! Invalid request code
    SMG$_FATERRLIB,         ! Fatal error in library
    SMG$_NO_ARGS;           ! No arguments to substitute

EXTERNAL
    SMG$$NUM_PARAMS : VECTOR [,BYTE];
                            ! number of parameters for each string cap
```

```
1 %SBTTL 'SMG$INIT_TERM_TABLE  - Initialize terminal table'
1 GLOBAL ROUTINE SMG$INIT_TERM_TABLE (
1                 P_TERM_NAME,
1                 P_TERM_TABLE      ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine points to a specific entry in the terminal capabilities
1 !       file, TERMTABLE.EXE.  TERM$TABLOC is searched first for a private
1 !       copy of TERMTABLE.EXE, and if this fails the system definition file
1 !       (in global section SMG$TERMTABLE) is used.  TERMTABLE.EXE must
1 !       previously have been created by compiling TERMTABLE.TXT.  The address
1 !       returned by this routine is used as an input to SMG$GET_TERM_DATA,
1 !       which fetches capability data.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$INIT_TERM_TABLE (P_TERM_NAME.rt.dx,
1 !                                               P_TERM_TABLE.wa.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       P_TERM_NAME             Name of the terminal - must be
1 !                               an entry in TERMTABLE.EXE
1 !
1 !       P_TERM_TABLE            Address of the table created
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       Contents of general terminal capabilities file.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SMG$_PRISECMAP          success - definition found in private TERMTABLE
1 !       SMG$_GBLSECMAP          success - definition found in global TERMTABLE
1 !       SMG$_UNDTERNAM          undefined terminal name
1 !       SMG$_UNDTERNOP          undefined terminal - no private terminal table
1 !       SMG$_UNDTERNOS          undefined terminal - no system terminal table
1 !       RMS errors
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
2     BEGIN
2     LOCAL
2       FOUND : INITIAL (1),            ! flag to indicate term def found
2       TMP_TERM_DEF,                   ! temp. place to store def addr
2       PREV_AST_STATE,                 ! retd by $SETAST to indicate AST
2                                       ! delivery was enabled or disabled
2       PRIV_STATUS : INITIAL (1),      ! status retd BY MAP_PRIV_SECTION
2       GBL_STATUS : INITIAL (1),       ! status retd by MAP_GBL_SECTION
```

```
        RET_STATUS : INITIAL (1);          ! status to return to caller

    $SMG$VALIDATE_ARGCOUNT (2,2);

    !+
    ! First we look for a private copy of TERMTABLE.EXE.  If
    ! PRIV_SECTION_ADDR is already set, then we don't need to map
    ! the section again.
    !-

    PREV_AST_STATE = $SETAST (ENBFLG = K_OFF);  ! disable ASTs while setting addr

    IF .PRIV_SECTION_ADDR EQL 0
    THEN
        PRIV_STATUS = MAP_PRIV_SECTION ();
                                            ! map priv section if possible

    IF .PREV_AST_STATE EQL SS$_WASSET
    THEN
        $SETAST (ENBFLG = K_ON);            ! re-enable ASTs


    !+
    ! See if the requested terminal definition is in the private section.
    !-

    IF .PRIV_SECTION_ADDR NEQ 0
    THEN
        BEGIN
        !+
        ! This routine will search the entire terminal index.
        !-
        FOUND = SEARCH_FOR_TERM_DEF (.PRIV_SECTION_ADDR, .P_TERM_NAME,
                                     TMP_TERM_DEF);
        RET_STATUS = SMG$_PRISECMAP;
        END;

    !+
    ! If there was no private copy of Termtable or the requested terminal definition
    ! wasn't defined there, then map to the global system copy.
    !-

    IF .FOUND NEQ K_MATCH
    THEN
        BEGIN

        PREV_AST_STATE = $SETAST (ENBFLG = K_OFF);      ! disable ASTs

        IF .GBL_SECTION_ADDR EQL 0
        THEN
            GBL_STATUS = MAP_GBL_SECTION ();

        IF .PREV_AST_STATE EQL SS$_WASSET
        THEN
            $SETAST (ENBFLG = K_ON);                    ! re-enable ASTs

        IF .GBL_SECTION_ADDR NEQ 0
```

```
;   3             THEN
;   4                 BEGIN
;   4                 FOUND = SEARCH_FOR_TERM_DEF (.GBL_SECTION_ADDR, .P_TERM_NAME,
;   4                                              TMP_TERM_DEF);
;   4                 RET_STATUS = SMG$_GBLSECMAP;
;   3                 END;
;   2             END;
;   2
;   2   !+
;   2   ! When we get here, we have looked in both the private and global copies
;   2   ! of Termtable.exe.  If we failed to find the requested terminal definition,
;   2   ! return an error.
;   2   !-
;   2
;   2         IF .FOUND NEQ K_MATCH
;   2         THEN
;   3             BEGIN
;   3             IF NOT .PRIV_STATUS
;   3             THEN
;   4                 RETURN (SMG$_UNDTERNOP)
;   3             ELSE
;   3                 IF NOT .GBL_STATUS
;   3                 THEN
;   4                     RETURN (SMG$_UNDTERNOS)
;   3             ELSE
;   4                 RETURN (SMG$_UNDTERNAM)
;   3             END
;   2         ELSE
;   2             .P_TERM_TABLE = .TMP_TERM_DEF;
;   2
;   2         SMG$$NUMBER_PARAMETERS ();              ! init expected # params for caps
;   2                                                 ! for future get_term_data calls
;   2
;   2         RETURN (.RET_STATUS);                   ! return success which also reveals
;   2                                                 ! where we found the defintition
;   2
;   1         END;                                    ! end of routine SMG$INIT_TERM_TABLE

                                                      .TITLE  SMG$INTERFACE_TERM_TABLE SMG$INTERFACE_TERM_TAB
;                                                                                     LE - User Interface to
;                                                                                      Te
                                                      .IDENT  \1-005\

                                                      .PSECT  _SMG$DATA,NOEXE,  PIC,2

                                   00000000  00000 GBL_SECTION_ADDR:
                                                      .LONG   0                                          ;
                                   00000000  00004 PRIV_SECTION_ADDR:
                                                      .LONG   0                                          ;
                                   00000000  00008 PRIV_SECTION_ADDR_END:
                                                      .LONG   0                                          ;
                                   00000000  0000C PRIV_SECTION_CHANNEL:
                                                      .LONG   0                                          ;
                                         FD  00010 K_OPERAND:
                                                      .BYTE   -3                                         ;
                                         FC  00011 K_SUBSTITUTE:
```

```
                                                       .BYTE   -4                                          ;
                        FB  00012 K_ADD:               .BYTE   -5                                          ;
                        FA  00013 K_SUBTRACT:
                                                       .BYTE   -6                                          ;
                        F9  00014 K_MULTIPLY:
                                                       .BYTE   -7                                          ;
                        F8  00015 K_DIVIDE:
                                                       .BYTE   -8                                          ;
                        F7  00016 K_STORE:             .BYTE   -9                                          ;

                                                       .EXTRN  SMG$$NUMBER_PARAMETERS
                                                       .EXTRN  STR$COPY_DX, STR$CASE_BLIND_COMPARE
                                                       .EXTRN  SMG$_WRONUMARG, SMG$_TABID_MIS
                                                       .EXTRN  SMG$_UNDTERNAM, SMG$_UNDTERTYP
                                                       .EXTRN  SMG$_PRISECMAP, SMG$_GBLSECMAP
                                                       .EXTRN  SMG$_UNDTERNOP, SMG$_UNDTERNOS
                                                       .EXTRN  SMG$_INVTERTAB, SMG$_INVREQCOD
                                                       .EXTRN  SMG$_FATERRLIB, SMG$_NO_ARGS
                                                       .EXTRN  SMG$$NUM_PARAMS
                                                       .EXTRN  SYS$SETAST

                                                       .PSECT  _SMG$CODE,NOWRT, SHR, PIC,2

                      01FC 00000                       .ENTRY  SMG$INIT_TERM_TABLE, Save R2,R3,R4,R5,R6,-  ; 0342
                                                               R7,R8                                       ;
          58 00000000G  00  9E 00002                   MOVAB   SYS$SETAST, R8                              ;
          57 00000000'  EF  9E 00009                   MOVAB   PRIV_SECTION_ADDR, R7                       ;
          5E            04  C2 00010                   SUBL2   #4, SP                                      ;
          54            01  D0 00013                   MOVL    #1, FOUND                                   ; 0390
          56            01  D0 00016                   MOVL    #1, PRIV_STATUS                             ;
          55            01  D0 00019                   MOVL    #1, GBL_STATUS                              ;
          52            01  D0 0001C                   MOVL    #1, RET_STATUS                              ;
          02            6C  91 0001F                   CMPB    (AP), #2                                    ; 0400
                        08  13 00022                   BEQL    1$                                          ;
          50 00000000G  8F  D0 00024                   MOVL    #SMG$_WRONUMARG, R0                         ;
                            04 0002B                   RET                                                 ;
                        7E  D4 0002C 1$:               CLRL    -(SP)                                       ; 0408
          68            01  FB 0002E                   CALLS   #1, SYS$SETAST                              ;
          53            50  D0 00031                   MOVL    R0, PREV_AST_STATE                          ;
                        67  D5 00034                   TSTL    PRIV_SECTION_ADDR                           ; 0410
                        08  12 00036                   BNEQ    2$                                          ;
   0000V  CF            00  FB 00038                   CALLS   #0, MAP_PRIV_SECTION                        ; 0412
          56            50  D0 0003D                   MOVL    R0, PRIV_STATUS                             ;
          09            53  D1 00040 2$:               CMPL    PREV_AST_STATE, #9                          ; 0415
                        05  12 00043                   BNEQ    3$                                          ;
                        01  DD 00045                   PUSHL   #1                                          ; 0417
          68            01  FB 00047                   CALLS   #1, SYS$SETAST                              ;
                        67  D5 0004A 3$:               TSTL    PRIV_SECTION_ADDR                           ; 0424
                        16  13 0004C                   BEQL    4$                                          ;
                        5E  DD 0004E                   PUSHL   SP                                          ; 0430
                  04    AC  DD 00050                   PUSHL   P_TERM_NAME                                 ;
                        67  DD 00053                   PUSHL   PRIV_SECTION_ADDR                           ;
   0000V  CF            03  FB 00055                   CALLS   #3, SEARCH_FOR_TERM_DEF                     ;
          54            50  D0 0005A                   MOVL    R0, FOUND                                   ;
          52 00000000G  00  9E 0005D                   MOVAB   SMG$_PRISECMAP, RET_STATUS                  ; 0432
                        54  D5 00064 4$:               TSTL    FOUND                                       ; 0440
                        3B  13 00066                   BEQL    7$                                          ;
```

```
                                7E  D4 00068         CLRL    -(SP)                                   ; 0444
                68              01  FB 0006A         CALLS   #1, SYS$SETAST                          ;
                53              50  D0 0006D         MOVL    R0, PREV_AST_STATE                      ;
                        FC      A7  D5 00070         TSTL    GBL_SECTION_ADDR                        ; 0446
                                08  12 00073         BNEQ    5$                                      ;
          0000V CF              00  FB 00075         CALLS   #0, MAP_GBL_SECTION                     ; 0448
                55              50  D0 0007A         MOVL    R0, GBL_STATUS                          ;
                09              53  D1 0007D 5$:     CMPL    PREV_AST_STATE, #9                      ; 0450
                                05  12 00080         BNEQ    6$                                      ;
                                01  DD 00082         PUSHL   #1                                      ; 0452
                68              01  FB 00084         CALLS   #1, SYS$SETAST                          ;
                        FC      A7  D5 00087 6$:     TSTL    GBL_SECTION_ADDR                        ; 0454
                                17  13 0008A         BEQL    7$                                      ;
                                5E  DD 0008C         PUSHL   SP                                      ; 0457
                        04      AC  DD 0008E         PUSHL   P_TERM_NAME                             ;
                        FC      A7  DD 00091         PUSHL   GBL_SECTION_ADDR                        ;
          0000V CF              03  FB 00094         CALLS   #3, SEARCH_FOR_TERM_DEF                 ;
                54              50  D0 00099         MOVL    R0, FOUND                               ;
                52 00000000G    00  9E 0009C         MOVAB   SMG$_GBLSECMAP, RET_STATUS              ; 0459
                                54  D5 000A3 7$:     TSTL    FOUND                                   ; 0469
                                1E  13 000A5         BEQL    10$                                     ;
                08              56  E8 000A7         BLBS    PRIV_STATUS, 8$                         ; 0472
                50 00000000G    00  9E 000AA         MOVAB   SMG$_UNDTERNOP, R0                      ; 0474
                                    04 000B1         RET                                             ; 0476
                08              55  E8 000B2 8$:     BLBS    GBL_STATUS, 9$                          ;
                50 00000000G    00  9E 000B5         MOVAB   SMG$_UNDTERNOS, R0                      ; 0478
                                    04 000BC         RET                                             ;
                50 00000000G    00  9E 000BD 9$:     MOVAB   SMG$_UNDTERNAM, R0                      ; 0480
                                    04 000C4         RET                                             ; 0476
             08 BC              6E  D0 000C5 10$:    MOVL    TMP_TERM_DEF, @P_TERM_TABLE             ; 0483
      00000000G 00              00  FB 000C9         CALLS   #0, SMG$$NUMBER_PARAMETERS              ; 0485
                50              52  D0 000D0         MOVL    RET_STATUS, R0                          ; 0488
                                    04 000D3         RET                                             ; 0491

; Routine Size:  212 bytes,    Routine Base:  _SMG$CODE + 0000
```

```
%SBTTL 'SMG$INIT_TERM_TABLE_BY_TYPE - Init terminal def address given type'
GLOBAL ROUTINE SMG$INIT_TERM_TABLE_BY_TYPE (
                                        P_TERM_TYPE,
                                        P_TERM_TABLE,
                                        P_TERM_NAME ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine points to a specific entry in the terminal capabilities
!       file, TERMTABLE.EXE.  TERM$TABLOC is searched first for a private
!       copy of TERMTABLE.EXE, and if this fails the system definition file
!       (in global section SMG$TERMTABLE) is used.  TERMTABLE.EXE must
!       previously have been created by compiling TERMTABLE.TXT.  The address
!       returned by this routine is used as an input to SMG$GET_TERM_DATA,
!       which fetches capability data.
!
!       This routine is similar to SMG$INIT_TERM_TABLE except that it
!       accepts a device type rather than a device name as input.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$INIT_TERM_TABLE_BY_TYPE (P_TERM_TYPE.rb.r,
!                                                        P_TERM_TABLE.wa.r
!                                                       [,P_TERM_NAME.wt.dx])
!
! FORMAL PARAMETERS:
!
!       P_TERM_TYPE             Device type of the terminal - must be
!                               an entry in TERMTABLE.EXE.  (The device
!                               type can be obtained via a call to $GETDVIW.)
!                               This is the address of a signed byte.
!
!       P_TERM_TABLE            Address of the table created
!
!       P_TERM_NAME             Optional descriptor to contain the name
!                               associated with this device type
!
! IMPLICIT INPUTS:
!
!       Contents of general terminal capabilities file.
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SMG$_PRISECMAP          success - definition found in private TERMTABLE
!       SMG$_GBLSECMAP          success - definition found in global TERMTABLE
!       SMG$_UNDTERTYP          undefined terminal type
!       SMG$_UNDTERNOP          undefined terminal - no private termtable
!       SMG$_UNDTERNOS          undefined terminal - no system termtable
!
! SIDE EFFECTS:
!
!       NONE
!--
```

```
2       BEGIN
2       BUILTIN
2           NULLPARAMETER;
2       LOCAL
2           FOUND : INITIAL (1),            ! flag to indicate term def found
2           TMP_TERM_DEF,                   ! temp. place to store def addr
2           TMP_TERM_NAME : REF VECTOR [,BYTE], ! temp. place to store name addr
2           PREV_AST_STATE,                 ! retd by $SETAST to indicate ASTs
2                                           ! previously enabled or disabled
2           PRIV_STATUS : INITIAL (1),      ! status retd by MAP_PRIV_SECTION
2           GBL_STATUS : INITIAL (1),       ! status retd by MAP_GBL_SECTION
2           RET_STATUS : INITIAL (1);
2       BIND
2           TERM_TYPE = .P_TERM_TYPE : SIGNED BYTE;
2
2       $SMG$VALIDATE_ARGCOUNT (2,3);
2
2       .P_TERM_TABLE = 0;                      ! init
2
2   !+
2   ! First we look for a private copy of TERMTABLE.EXE.  If
2   ! PRIV_SECTION_ADDR is already set, then we don't need to map
2   ! the section again.
2   !-
2
2       PREV_AST_STATE = $SETAST (ENBFLG = K_OFF);  ! disable ASTs while setting addr
2
2       IF .PRIV_SECTION_ADDR EQL 0
2       THEN
2           PRIV_STATUS = MAP_PRIV_SECTION ();      ! map priv section if possible
2
2       IF .PREV_AST_STATE EQL SS$_WASSET
2       THEN
2           $SETAST (ENBFLG = K_ON);                ! re-enable ASTs
2
2
2   !+
2   ! See if the requested terminal definition is in the private section.
2   ! If this terminal type number is negative but not in the private range, skip this
2   ! and proceed to check the global table.
2   !-
2
2       IF .PRIV_SECTION_ADDR NEQ 0 AND
3          (.TERM_TYPE LSS K_PRIV_TYPE  OR .TERM_TYPE GEQ 0)
2       THEN
3           BEGIN
3           !+
3           ! This routine will search the entire terminal index.
3           !-
3           FOUND = SEARCH_FOR_TERM_DEF_BY_TYPE (.PRIV_SECTION_ADDR, .TERM_TYPE,
3                                               TMP_TERM_DEF, TMP_TERM_NAME);
3           RET_STATUS = SMG$_PRISECMAP;
2           END;
2
2   !+
2   ! If there was no private copy of Termtable or the requested terminal definition
2   ! wasn't defined there, then map to the global system copy.
```

```
!
! To prevent private and global Termtable definitions from having the same
! terminal type numbers assigned, we always bias private ones by -65.  No
! need to check the global definition for a private number.
! Positive types and negative types larger than or equal to K_PRIV_TYPE are ok.
!-

    IF .FOUND NEQ K_MATCH AND
       .TERM_TYPE GEQ K_PRIV_TYPE
    THEN
        BEGIN

        PREV_AST_STATE = $SETAST (ENBFLG = K_OFF);       ! disable ASTs

        IF .GBL_SECTION_ADDR EQL 0
        THEN
            GBL_STATUS = MAP_GBL_SECTION ();

        IF .PREV_AST_STATE EQL SS$_WASSET
        THEN
            $SETAST (ENBFLG = K_ON);                     ! re-enable ASTs

        IF .GBL_SECTION_ADDR NEQ 0
        THEN
            BEGIN
            FOUND = SEARCH_FOR_TERM_DEF_BY_TYPE (.GBL_SECTION_ADDR,
                                                 .TERM_TYPE,
                                                 TMP_TERM_DEF,
                                                 TMP_TERM_NAME);
            RET_STATUS = SMG$_GBLSECMAP;
            END;
        END;

!+
! When we get here, we have looked in both the private and global copies
! of Termtable.exe.  If we failed to find the requested terminal definition,
! return an error.
!-

    IF .FOUND NEQ K_MATCH
    THEN
        BEGIN
        IF NOT .PRIV_STATUS
        THEN
            RETURN (SMG$_UNDTERNOP)
        ELSE
            IF NOT .GBL_STATUS
            THEN
                RETURN (SMG$_UNDTERNOS)
        ELSE
            RETURN (SMG$_UNDTERNAM)
        END
    ELSE
        BEGIN
        .P_TERM_TABLE = .TMP_TERM_DEF;
                                        ! return addr of terminal definition
        IF NOT NULLPARAMETER (P_TERM_NAME)
```

D 2

```
;  589      0663  3           THEN
;  590      0664  4               BEGIN                         ! return term name if requested
;  591      0665  4               LOCAL
;  592      0666  4                   TERM_NAME_DESC : BLOCK [8,BYTE];
;  593      0667  4
;  594      0668  4               TERM_NAME_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  595      0669  4               TERM_NAME_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  596      0670  4               TERM_NAME_DESC [DSC$W_LENGTH] = .TMP_TERM_NAME [0];
;  597      0671  4               TERM_NAME_DESC [DSC$A_POINTER] = TMP_TERM_NAME [2];
;  598      0672  4
;  599      0673  4               STR$COPY_DX (.P_TERM_NAME, TERM_NAME_DESC);
;  600      0674  3               END;
;  601      0675  2           END;
;  602      0676  2
;  603      0677  2       SMG$$NUMBER_PARAMETERS ();          ! init expected # params for caps
;  604      0678  2                                           ! for future get_term_data calls
;  605      0679  2
;  606      0680  2       RETURN (.RET_STATUS);           ! success indicating where we found
;  607      0681  2                                       ! the definition
;  608      0682  2
;  609      0683  1       END;                            ! end of routine SMG$INIT_TERM_TABLE_BY_TYPE



                                   01FC 00000        .ENTRY  SMG$INIT_TERM_TABLE_BY_TYPE, Save R2,R3,R4,-; 0493
                                                             R5,R6,R7,R8                                   ;
                 58 00000000G  00   9E 00002        MOVAB   SYS$SETAST, R8                                ;
                 57 000000C0'  EF   9E 00009        MOVAB   PRIV_SECTION_ADDR, R7                         ;
                 5E            10   C2 00010        SUBL2   #16, SP                                       ;
                 53            01   D0 00013        MOVL    #1, FOUND                                     ; 0549
                 56            01   D0 00016        MOVL    #1, PRIV_STATUS                               ;
                 55            01   D0 00019        MOVL    #1, GBL_STATUS                                ;
                 54            01   D0 0001C        MOVL    #1, RET_STATUS                                ;
        50       6C            02   83 0001F        SUBB3   #2, (AP), DIFF                                ; 0564
                 01            50   91 00023        CMPB    DIFF, #1                                      ;
                               08   1B 00026        BLEQU   1$                                            ;
                 50 00000000G  8F   D0 00028        MOVL    #SMG$_WRONUMARG, R0                           ;
                               04   0002F           RET                                                   ;
                         08    BC   D4 00030 1$:    CLRL    @P_TERM_TABLE                                 ; 0566
                               7E   D4 00033        CLRL    -(SP)                                         ; 0574
                 68            01   FB 00035        CALLS   #1, SYS$SETAST                                ;
                 52            50   D0 00038        MOVL    R0, PREV_AST_STATE                            ;
                               67   D5 0003B        TSTL    PRIV_SECTION_ADDR                             ; 0576
                               08   12 0003D        BNEQ    2$                                            ;
           0000V CF            00   FB 0003F        CALLS   #0, MAP_PRIV_SECTION                          ; 0578
                 56            50   D0 00044        MOVL    R0, PRIV_STATUS                               ;
                 09            52   D1 00047 2$:    CMPL    PREV_AST_STATE, #9                            ; 0580
                               05   12 0004A        BNEQ    3$                                            ;
                               01   DD 0004C        PUSHL   #1                                            ; 0582
                 68            01   FB 0004E        CALLS   #1, SYS$SETAST                                ;
                               67   D5 00051 3$:    TSTL    PRIV_SECTION_ADDR                             ; 0591
                               26   13 00053        BEQL    5$                                            ;
              BF 8F          04 BC  91 00055        CMPB    @P_TERM_TYPE, #-65                            ; 0592
                               05   19 0005A        BLSS    4$                                            ;
                             04 BC  95 0005C        TSTB    @P_TERM_TYPE                                  ;
```

```
                        1A  19 0005F       BLSS    5$                                        ;
                        5E  DD 00061 4$:   PUSHL   SP                                        ; 0598
                  08    AE  9F 00063       PUSHAB  TMP_TERM_DEF                              ;
            7E    04    BC  98 00066       CVTBL   @P_TERM_TYPE, -(SP)                       ;
                        67  DD 0006A       PUSHL   PRIV_SECTION_ADDR                         ;
       0000V  CF        04  FB 0006C       CALLS   #4, SEARCH_FOR_TERM_DEF_BY_TYPE           ;
              53        50  D0 00071       MOVL    R0, FOUND                                 ;
              54 00000000G  00  9E 00074   MOVAB   SMG$_PRISECMAP, RET_STATUS                ; 0600
                        53  D5 0007B 5$:   TSTL    FOUND                                     ; 0613
                        46  13 0007D       BEQL    8$                                        ;
          BF  8F    04  BC  91 0007F       CMPB    @P_TERM_TYPE, #-65                        ; 0614
                        3F  19 00084       BLSS    8$                                        ;
                        7E  D4 00086       CLRL    -(SP)                                     ; 0618
              68        01  FB 00088       CALLS   #1, SYS$SETAST                            ;
              52        50  D0 0008B       MOVL    R0, PREV_AST_STATE                        ;
                  FC    A7  D5 0008E       TSTL    GBL_SECTION_ADDR                          ; 0620
                        08  12 00091       BNEQ    6$                                        ;
       0000V  CF        00  FB 00093       CALLS   #0, MAP_GBL_SECTION                       ; 0622
              55        50  D0 00098       MOVL    R0, GBL_STATUS                            ;
              09        52  D1 0009B 6$:   CMPL    PREV_AST_STATE, #9                        ; 0624
                        05  12 0009E       BNEQ    7$                                        ;
                        01  DD 000A0       PUSHL   #1                                        ; 0626
              68        01  FB 000A2       CALLS   #1, SYS$SETAST                            ;
                  FC    A7  D5 000A5 7$:   TSTL    GBL_SECTION_ADDR                          ; 0628
                        1B  13 000A8       BEQL    8$                                        ;
                        5E  DD 000AA       PUSHL   SP                                        ; 0631
                  08    AE  9F 000AC       PUSHAB  TMP_TERM_DEF                              ;
            7E    04    BC  98 000AF       CVTBL   @P_TERM_TYPE, -(SP)                       ; 0632
                  FC    A7  DD 000B3       PUSHL   GBL_SECTION_ADDR                          ; 0631
       0000V  CF        04  FB 000B6       CALLS   #4, SEARCH_FOR_TERM_DEF_BY_TYPE           ;
              53        50  D0 000BB       MOVL    R0, FOUND                                 ;
              54 00000000G  00  9E 000BE   MOVAB   SMG$_GBLSECMAP, RET_STATUS                ; 0635
                        53  D5 000C5 8$:   TSTL    FOUND                                     ; 0645
                        1E  13 000C7       BEQL    11$                                       ;
              08        56  E8 000C9       BLBS    PRIV_STATUS, 9$                           ; 0648
              50 00000000G  00  9E 000CC   MOVAB   SMG$_UNDTERNOP, R0                        ; 0650
                            04 000D3       RET                                               ; 0652
              08        55  E8 000D4 9$:   BLBS    GBL_STATUS, 10$                           ;
              50 00000000G  00  9E 000D7   MOVAB   SMG$_UNDTERNOS, R0                        ; 0654
                            04 000DE       RET                                               ;
              50 00000000G  00  9E 000DF 10$: MOVAB SMG$_UNDTERNAM, R0                       ; 0656
                            04 000E6       RET                                               ; 0652
          08  BC    04  AE  D0 000E7 11$:  MOVL    TMP_TERM_DEF, @P_TERM_TABLE               ; 0660
              03        6C  91 000EC       CMPB    (AP), #3                                  ; 0662
                        22  1F 000EF       BLSSU   12$                                       ;
                  0C    AC  D5 000F1       TSTL    12(AP)                                    ;
                        1D  13 000F4       BEQL    12$                                       ;
          0A  AE  010E  8F  B0 000F6       MOVW    #270, TERM_NAME_DESC+2                    ; 0668
          08  AE    00  BE  9B 000FC       MOVZBW  @TMP_TERM_NAME, TERM_NAME_DESC            ; 0670
    0C    AE        6E  02  C1 00101       ADDL3   #2, TMP_TERM_NAME, TERM_NAME_DESC+4       ; 0671
                  08    AE  9F 00106       PUSHAB  TERM_NAME_DESC                            ; 0673
                  0C    AC  DD 00109       PUSHL   P_TERM_NAME                               ;
   00000000G  00        02  FB 0010C       CALLS   #2, STR$COPY_DX                           ;
   00000000G  00        00  FB 00113 12$:  CALLS   #0, SMG$$NUMBER_PARAMETERS                ; 0677
              50        54  D0 0011A       MOVL    RET_STATUS, R0                            ; 0680
                            04 0011D       RET                                               ; 0683
```

```
; Routine Size:  286 bytes,    Routine Base:  _SMG$CODE + 00D4
```

```
1 %SBTTL 'SMG$GET_TERM_DATA  - Get terminal data'
1 GLOBAL ROUTINE SMG$GET_TERM_DATA (
1                                   P_TERM_TABLE,
1                                   REQUEST_CODE,
1                                   MAX_BUFFER_LENGTH,
1                                   RETURN_LENGTH,
1                                   BUFFER_ADDRESS,
1                                   INPUT_ARG_VECTOR : REF VECTOR [,LONG]
1                                   ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine extracts the requested information from
1 !       the specified terminal table.  The terminal table address must
1 !       have been obtained by a call to SMG$INIT_TERM_TABLE or
1 !       SMG$INIT_TERM_TABLE_BY_TYPE prior to calling this routine.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$GET_TERM_DATA (P_TERM_TABLE.ra.r,
1 !                                             REQUEST_CODE.rl.r,
1 !                                             MAX_BUFFER_LENGTH.rl.r,
1 !                                             RETURN_LENGTH.wl.r,
1 !                                             BUFFER_ADDRESS.wa.r
1 !                                             [,INPUT_ARG_VECTOR.rl.r]
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       P_TERM_TABLE            Address of terminal capabilities table
1 !                               for desired terminal type.  (See
1 !                               SMG$INIT_TERM_TABLE to obtain this address.)
1 !
1 !       REQUEST_CODE            Longword constant of the form
1 !                               SMG$K_xyz (defined in Digital-supplied
1 !                               library).  Specifies desired capability.
1 !                               Xyz corresponds to a keyword in the terminal
1 !                               definition (for instance ANSI_CRT).
1 !
1 !       MAX_BUFFER_LENGTH       Longword number of bytes of the buffer
1 !                               in which the requested capability data will be
1 !                               returned.  Most capabilities will fit in
1 !                               20 bytes or less - you will probably want
1 !                               to allocate a little extra.
1 !
1 !       RETURN_LENGTH           Longword length of the data returned.
1 !                               It indicates how many bytes of the buffer are
1 !                               actually used.
1 !
1 !       BUFFER_ADDRESS          Longword address of the buffer in which
1 !                               to return the capability data.
1 !
1 !       INPUT_ARG_VECTOR        Optional argument used only for a
1 !                               'variable' capability.  Address of a longword
1 !                               vector.  The number of longwords which follow
1 !                               must be in the first longword.  This vector
1 !                               is used with capability string containing
```

```
!                                       ! or %.  You may provide the correct number of
!                                       values to be used with each ! and % in the
!                                       string or you may default all of the values.
!                                       SMG$GET_TERM_DATA will supply a default value
!                                       of 1.
!
!
! IMPLICIT INPUTS:
!
!       Contents of system terminal capabilities file.
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SMG$_INVTERTAB          invalid terminal table address
!       SMG$_INVREQCOD          invalid request code
!       SS$_NORMAL              success
!
! SIDE EFFECTS:
!
!       NONE
!--
    BEGIN
    BUILTIN
        NULLPARAMETER;
    LOCAL
        DATA : REF VECTOR [,BYTE,SIGNED], ! terminal data
        DATA_LENGTH;                    ! length of terminal data
    BIND
        TERM_DEF = ..P_TERM_TABLE,      ! start of terminal definition
        TERM_DEF_PTRS = TERM_DEF : VECTOR [,WORD],
                                        ! start of data pointers
        START_DEF_DATA = TERM_DEF_PTRS + SMG$K_CAP_PTRS_SIZE;
                                        ! start of actual data

    $SMG$VALIDATE_ARGCOUNT (5, 6);

!+
! Validate that we have a valid TERMTABLE.  The terminal
! definition should contain an id in the first byte.
!-

    IF .TERM_DEF <0,8> NEQ SMG$K_TERM_DEF_ID
    THEN
        RETURN (SMG$_INVTERTAB);

!+
! Make sure this is a valid request code.
!-

    IF ..REQUEST_CODE LSS SMG$K_MIN_REQUEST_CODE OR
       ..REQUEST_CODE GTR SMG$K_MAX_REQUEST_CODE
    THEN
        RETURN (SMG$_INVREQCOD);
```

```
2
2  !+
2  ! Find the requested terminal data.
2  !-
2
2      .RETURN_LENGTH = 0;                     ! init
2
2      IF .TERM_DEF_PTRS [..REQUEST_CODE] EQL 0
2      THEN
2          RETURN (SS$_NORMAL);
2
2      DATA = .TERM_DEF_PTRS [..REQUEST_CODE] + TERM_DEF;
2                                      ! offset + start of data area
2      DATA_LENGTH = .DATA [0];
2
2      IF .DATA_LENGTH EQL 0
2      THEN
2          RETURN (SS$_NORMAL);        ! requested data not available
2
2  !+
2  ! Branch on whether additional processing is required.
2  !-
2
2      IF .DATA_LENGTH GTR 0
2      THEN
3          BEGIN                       ! static data - copy into user's buffer
3          LOCAL
3              RET_LEN;
3
3          RET_LEN = MIN (..MAX_BUFFER_LENGTH, .DATA_LENGTH);
3
3          IF ..REQUEST_CODE EQL SMG$K_VMS_TERMINAL_NUMBER
3          THEN
4              $BIAS_TYPE_NUMBER (..P_TERM_TABLE, .DATA [2], .RET_LEN, .BUFFER_ADDRESS)
4                                      ! private numbers must be -65 to -127
3          ELSE
3              CH$MOVE (.RET_LEN, DATA [2], .BUFFER_ADDRESS);
3          !+
3          ! Return length to caller.
3          !-
3          IF .RET_LEN NEQ 0
3          THEN
3              .RETURN_LENGTH = .RET_LEN;
3          END                         ! static data - copy into user's buffer
3
2      ELSE
2
3          BEGIN                       ! dynamic - additional processing needed
3          LOCAL
3              LOCAL_ARGS : VECTOR [10,LONG] INITIAL (REP 10 OF (1)),
3              KIND;
3          KIND = .DATA [1];                   ! make this a longword for comparison
3
3          !+
3          ! Use either the defaults or the parameters passed by the caller.
3          !-
3          $CHECK_SUFFICIENT_USER_ARGS;
```

```
		IF .KIND EQL SMG$K_FAO_STRING
		THEN
		    BEGIN				! FAO processing
		    !+
		    ! Process FAO string.
		    !-
		    LOCAL
			FAO_STATUS,
			CONTROL_DSC : BLOCK [8,BYTE],
			RETURN_DSC : BLOCK [8,BYTE];

		    CONTROL_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
		    CONTROL_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
		    CONTROL_DSC [DSC$W_LENGTH] = ABS (.DATA_LENGTH);
		    CONTROL_DSC [DSC$A_POINTER] = DATA [2];

		    RETURN_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
		    RETURN_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
		    RETURN_DSC [DSC$W_LENGTH] = ..MAX_BUFFER_LENGTH;
		    RETURN_DSC [DSC$A_POINTER] = .BUFFER_ADDRESS;

		    FAO_STATUS = $FAOL (CTRSTR = CONTROL_DSC, OUTLEN = .RETURN_LENGTH,
				    OUTBUF = RETURN_DSC, PRMLST = LOCAL_ARGS [0]);
		    IF NOT .FAO_STATUS
		    THEN
			RETURN (.FAO_STATUS);

		    END				! end of FAO processing

		ELSE

		IF .KIND EQL SMG$K_ARITH_STRING
		THEN
		    BEGIN					! arithmetic processing
		    LOCAL
			RET_LEN,
			DECODED_ARITH_STRING : BLOCK [255,BYTE],
			DECODED_LENGTH,
			STATUS;

		    IF NOT (STATUS = DECODE_ARITH_STRING (DATA [0],
							LOCAL_ARGS [0],
							 DECODED_ARITH_STRING,
							 DECODED_LENGTH))
		    THEN
			RETURN (.STATUS);

		    RET_LEN = MIN (..MAX_BUFFER_LENGTH, .DECODED_LENGTH);
		    CH$MOVE (.RET_LEN, DECODED_ARITH_STRING, .BUFFER_ADDRESS);
						! copy decoded stuff to user buffer
		    IF .RET_LEN NEQ 0
		    THEN
			.RETURN_LENGTH = .RET_LEN;
						! return length to caller
		    END				! end of arithmetic processing
```

```
;    839     0912  3          ELSE
;    840     0913  3              RETURN (SMG$_FATERRLIB);     ! unknown type code
;    841     0914  3
;    842     0915  2          END;                             ! dynamic - additional processing needed
;    843     0916  2
;    844     0917  2      RETURN (SS$_NORMAL);
;    845     0918  1      END;                                 ! end of SMG$GET_TERM_DATA


                                              001F2           .BLKB   2
                                  00000001#   001F4 P.AAA:    .LONG   1[10]                                     ;

                                                              .EXTRN  SYS$FAOL

                                        07FC  00000           .ENTRY  SMG$GET_TERM_DATA, Save R2,R3,R4,R5,R6,R7,-   ; 0685
                                                                      R8,R9,R10                                     ;
                 5A 00000000'  EF   9E  00002           MOVAB   PRIV_SECTION_ADDR, R10                        ;
                 5E      FED0  CE   9E  00009           MOVAB   -304(SP), SP                                  ;
                 51        04  BC   D0  0000E           MOVL    @P_TERM_TABLE, R1                             ; 0773
       50        6C            05   83  00012           SUBB3   #5, (AP), DIFF                                ; 0779
                 01            50   91  00016           CMPB    DIFF, #1                                      ;
                               08   1B  00019           BLEQU   1$                                            ;
                 50 00000000G  8F   D0  0001B           MOVL    #SMG$_WRONUMARG, R0                           ;
                                    04  00022           RET                                                   ;
            89   8F            61   91  00023 1$:       CMPB    (R1), #137                                    ; 0786
                               08   13  00027           BEQL    2$                                            ;
                 50 00000000G  00   9E  00029           MOVAB   SMG$_INVTERTAB, R0                            ; 0788
                                    04  00030           RET                                                   ;
                 57        08  BC   D0  00031 2$:       MOVL    @REQUEST_CODE, R7                             ; 0794
                               09   15  00035           BLEQ    3$                                            ;
       00000294  8F            57   D1  00037           CMPL    R7, #660                                      ; 0795
                               08   15  0003E           BLEQ    4$                                            ;
                 50 00000000G  00   9E  00040 3$:       MOVAB   SMG$_INVREQCOD, R0                            ; 0797
                                    04  00047           RET                                                   ;
                           10  BC   D4  00048 4$:       CLRL    @RETURN_LENGTH                                ; 0803
                             6147   B5  0004B           TSTW    (R1)[R7]                                      ; 0805
                               55   13  0004E           BEQL    9$                                            ;
                 58          6147   3C  00050           MOVZWL  (R1)[R7], DATA                                ; 0809
                 58            51   C0  00054           ADDL2   R1, DATA                                      ;
                 56            68   98  00057           CVTBL   (DATA), DATA_LENGTH                           ; 0811
                               49   13  0005A           BEQL    9$                                            ; 0813
                               4A   15  0005C           BLEQ    10$                                           ; 0821
                 50        0C  BC   D0  0005E           MOVL    @MAX_BUFFER_LENGTH, R0                        ; 0827
                 56            50   D1  00062           CMPL    R0, DATA_LENGTH                               ;
                               03   15  00065           BLEQ    5$                                            ;
                 50            56   D0  00067           MOVL    DATA_LENGTH, R0                               ;
                 59            50   D0  0006A 5$:       MOVL    R0, RET_LEN                                   ;
       000000E3  8F            57   D1  0006D           CMPL    R7, #227                                      ; 0829
                               21   12  00074           BNEQ    7$                                            ;
                 6E        02  A8   98  00076           CVTBL   2(DATA), BIAS_TYPE                            ; 0831
                               6A   D5  0007A           TSTL    PRIV_SECTION_ADDR                             ;
                               12   13  0007C           BEQL    6$                                            ;
                 6A        04  BC   D1  0007E           CMPL    @P_TERM_TABLE, PRIV_SECTION_ADDR              ;
                               0C   19  00082           BLSS    6$                                            ;
                           02  A8   95  00084           TSTB    2(DATA)                                       ;
                               07   18  00087           BGEQ    6$                                            ;
```

```
                  6E 00000041    8F  C2 00089        SUBL2   #65, BIAS_TYPE                                   :
      14   BC     6E             59  28 00090 6$:    MOVC3   RET_LEN, BIAS_TYPE, @BUFFER_ADDRESS              :
                                 06  11 00095        BRB     8$                                               : 0829
      14   BC     02   A8        59  28 00097 7$:    MOVC3   RET_LEN, 2(DATA), @BUFFER_ADDRESS                : 0834
                                 59  D5 0009D 8$:    TSTL    RET_LEN                                          : 0838
                                 04  13 0009F        BEQL    9$                                               :
           10     BC             59  D0 000A1        MOVL    RET_LEN, @RETURN_LENGTH                          : 0840
                               00CF  31 000A5 9$:    BRW     19$                                              : 0821
      D8   AD   FF2B   CF        28  28 000A8 10$:   MOVC3   #40, P.AAA, LOCAL_ARGS                           : 0847
                       53    01  A8  98 000AF        CVTBL   1(DATA), KIND                                    : 0849
                       06        6C  91 000B3        CMPB    (AP), #6                                         :
                                 34  1F 000B6        BLSSU   14$                                              :
                             18  AC  D5 000B8        TSTL    24(AP)                                           :
                                 2F  13 000BB        BEQL    14$                                              :
                             18  BC  D5 000BD        TSTL    @INPUT_ARG_VECTOR                                :
                                 2A  13 000C0        BEQL    14$                                              :
                  52 00000000G0047   9A 000C2        MOVZBL  SMG$$NUM_PARAMS[R7], R2                          :
                  52         18  BC  D1 000CA        CMPL    @INPUT_ARG_VECTOR, R2                            :
                                 08  13 000CE        BEQL    11$                                              :
                  50 00000000G   00  9E 000D0        MOVAB   SMG$_NO_ARGS, R0                                 :
                                     04 000D7        RET                                                      :
                  50             01  CE 000D8 11$:   MNEGL   #1, CTR                                          :
                                 0B  11 000DB        BRB     13$                                              :
                  51        18 BC40  DE 000DD 12$:   MOVAL   @INPUT_ARG_VECTOR[CTR], R1                       :
               D8 AD40       04  A1  D0 000E2        MOVL    4(R1), LOCAL_ARGS[CTR]                           :
           F1     50             52  F2 000E8 13$:   AOBLSS  R2, CTR, 12$                                     :
        FFFFFFFF  8F             53  D1 000EC 14$:   CMPL    KIND, #-1                                        : 0856
                                 3D  12 000F3        BNEQ    16$                                              :
      D2   AD   010E             8F  B0 000F5        MOVW    #270, CONTROL_DSC+2                              : 0867
                                 56  D5 000FB        TSTL    R6                                               : 0869
                                 03  18 000FD        BGEQ    15$                                              :
                  56             56  CE 000FF        MNEGL   R6, R6                                           :
                  D0  AD         56  B0 00102 15$:   MOVW    R6, CONTROL_DSC                                  :
                  D4  AD     02  A8  9E 00106        MOVAB   2(R8), CONTROL_DSC+4                             : 0870
                  CA  AD   010E  8F  B0 0010B        MOVW    #270, RETURN_DSC+2                               : 0872
                  C8  AD     0C  BC  B0 00111        MOVW    @MAX_BUFFER_LENGTH, RETURN_DSC                   : 0874
                  CC  AD     14  AC  D0 00116        MOVL    BUFFER_ADDRESS, RETURN_DSC+4                     : 0875
                             D8  AD  9F 0011B        PUSHAB  LOCAL_ARGS                                       : 0878
                             C8  AD  9F 0011E        PUSHAB  RETURN_DSC                                       :
                             10  AC  DD 00121        PUSHL   RETURN_LENGTH                                    :
                             D0  AD  9F 00124        PUSHAB  CONTROL_DSC                                      :
        00000000G 00             04  FB 00127        CALLS   #4, SYS$FAOL                                     :
                  46             50  E8 0012E        BLBS    FAO_STATUS, 19$                                  : 0879
                                     04 00131        RET                                                      : 0881
        FFFFFFFE  8F             53  D1 00132 16$:   CMPL    KIND, #-2                                        : 0887
                                 34  12 00139        BNEQ    18$                                              :
                             04  AE  9F 0013B        PUSHAB  DECODED_LENGTH                                   : 0896
                             0C  AE  9F 0013E        PUSHAB  DECODED_ARITH_STRING                             :
                             D8  AD  9F 00141        PUSHAB  LOCAL_ARGS                                       : 0897
                                 58  DD 00144        PUSHL   DATA                                             : 0896
            0000V CF             04  FB 00146        CALLS   #4, DECODE_ARITH_STRING                          :
                  2C             50  E9 0014B        BLBC    STATUS, 20$                                      :
                  50         0C  BC  D0 0014E        MOVL    @MAX_BUFFER_LENGTH, R0                           : 0903
           04     AE             50  D1 00152        CMPL    R0, DECODED_LENGTH                               :
                                 04  15 00156        BLEQ    17$                                              :
                  50         04  AE  D0 00158        MOVL    DECODED_LENGTH, R0                               :
                  56             50  D0 0015C 17$:   MOVL    R0, RET_LEN                                      :
```

```
          14   BC        08   AE             56  28 0015F         MOVC3   RET_LEN, DECODED_ARITH_STRING, -          ; 0904
                                                                          @BUFFER_ADDRESS                            ;
                                             56  D5 00165         TSTL    RET_LEN                                    ; 0906
                                             0E  13 00167         BEQL    19$                                        ;
                         10   BC             56  D0 00169         MOVL    RET_LEN, @RETURN_LENGTH                    ; 0908
                                             08  11 0016D         BRB     19$                                        ; 0887
                              50 00000000G   00  9E 0016F 18$:    MOVAB   SMG$_FATERRLIB, R0                         ; 0913
                                                 04 00176         RET                                                ;
                              50             01  D0 00177 19$:    MOVL    #1, R0                                     ; 0917
                                                 04 0017A 20$:    RET                                                ; 0918

; Routine Size:  379 bytes,    Routine Base:  _SMG$CODE + 021C
```

SMG
1-0

```
1 %SBTTL 'SMG$DEL_TERM_TABLE  - Delete terminal table'
1 GLOBAL ROUTINE SMG$DEL_TERM_TABLE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine terminates access to TERMTABLE.EXE.
1 !
1 !       If a private TERMTABLE.EXE was used, then we created a
1 !       temporary section to map it into the user's virtual address
1 !       space.  This routine will free up that virtual space so that
1 !       the user can re-use it.
1 !
1 !       If the system TERMTABLE.EXE was used, then we used the system
1 !       global section, and we do not have to free any virtual address
1 !       space.  In this case we simply return SS$_NORMAL.
1 !
1 !       In both cases (temporary and global section), this call is
1 !       optional.  Even for a temporary section, the virtual memory
1 !       would be automatically freed when the image terminated.
1 !       This routine is useful only in the case where a caller may
1 !       need to re-use the virtual address space.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$DEL_TERM_TABLE
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
2     BEGIN
2     LOCAL
2         PREV_AST_STATE;
2
2 !+
2 ! If the terminal definition came from the system global section,
2 ! then we simply return.  If the definition came from a private
2 ! temporary section, then we free up the virtual address space.
2 !-
2
```

```
;  904    0976  2      PREV_AST_STATE = $SETAST (ENBFLG = K_OFF);  ! disable ASTs
;  905    0977  2
;  906    0978  2      IF .PRIV_SECTION_ADDR NEQ 0
;  907    0979  2      THEN
;  908    0980  3          BEGIN                   ! free va - temp section
;  909    0981  3          LOCAL
;  910    0982  3              FREE_ADDR : VECTOR [2],
;  911    0983  3              STATUS;
;  912    0984  3
;  913    0985  3          FREE_ADDR [0] = .PRIV_SECTION_ADDR;
;  914    0986  3          FREE_ADDR [1] = .PRIV_SECTION_ADDR_END;
;  915    0987  4          IF NOT (STATUS = $DELTVA (INADR = FREE_ADDR))
;  916    0988  3          THEN
;  917    0989  3              RETURN (.STATUS);
;  918    0990  3
;  919    0991  4          IF NOT (STATUS = $DASSGN (CHAN = .PRIV_SECTION_CHANNEL))
;  920    0992  3          THEN
;  921    0993  3              RETURN (.STATUS);
;  922    0994  3
;  923    0995  3          PRIV_SECTION_ADDR = 0;
;  924    0996  3          PRIV_SECTION_CHANNEL = 0;
;  925    0997  2          END;                            ! free va - temp section
;  926    0998  2
;  927    0999  2      IF .PREV_AST_STATE EQL SS$_WASSET
;  928    1000  2      THEN
;  929    1001  2          $SETAST (ENBFLG = K_ON);                   ! re-enable ASTs
;  930    1002  2
;  931    1003  2      RETURN (SS$_NORMAL);
;  932    1004  1      END;                            ! end of SMG$DEL_TERM_TABLE


                                                     .EXTRN   SYS$DELTVA, SYS$DASSGN

                                      001C 00000     .ENTRY   SMG$DEL_TERM_TABLE, Save R2,R3,R4      ; 0920
                  54 00000000G    00  9E 00002      MOVAB    SYS$SETAST, R4                          ;
                  53 00000000'    EF  9E 00009      MOVAB    PRIV_SECTION_ADDR, R3                   ;
                  5E              08  C2 00010      SUBL2    #8, SP                                  ;
                                  7E  D4 00013      CLRL     -(SP)                                   ; 0976
                  64              01  FB 00015      CALLS    #1, SYS$SETAST                          ;
                  52              50  D0 00018      MOVL     R0, PREV_AST_STATE                      ;
                                  63  D5 0001B      TSTL     PRIV_SECTION_ADDR                       ; 0978
                                  29  13 0001D      BEQL     1$                                      ;
                  6E              63  D0 0001F      MOVL     PRIV_SECTION_ADDR, FREE_ADDR            ; 0985
           04     AE        04    A3  D0 00022      MOVL     PRIV_SECTION_ADDR_END, FREE_ADDR+4      ; 0986
                                  7E  7C 00027      CLRQ     -(SP)                                   ; 0987
                            08    AE  9F 00029      PUSHAB   FREE_ADDR                               ;
          00000000G       00      03  FB 0002C      CALLS    #3, SYS$DELTVA                          ;
                          1F      50  E9 00033      BLBC     STATUS, 3$                              ;
                            08    A3  DD 00036      PUSHL    PRIV_SECTION_CHANNEL                    ; 0991
          00000000G       00      01  FB 00039      CALLS    #1, SYS$DASSGN                          ;
                          12      50  E9 00040      BLBC     STATUS, 3$                              ;
                                  63  D4 00043      CLRL     PRIV_SECTION_ADDR                       ; 0995
                            08    A3  D4 00045      CLRL     PRIV_SECTION_CHANNEL                    ; 0996
                  09              52  D1 00048 1$:  CMPL     PREV_AST_STATE, #9                      ; 0999
                                  05  12 0004B      BNEQ     2$                                      ;
                                  01  DD 0004D      PUSHL    #1                                      ; 1001
```

```
          64      01 FB 0004F        CALLS   #1, SYS$SETAST                  ;
          50      01 D0 00052 2$:    MOVL    #1, R0                          ; 1003
                     04 00055 3$:    RET                                     ; 1004

; Routine Size:  86 bytes,    Routine Base:  _SMG$CODE + 0397
```

SMG
1-00
; Ro

```
1 %SBTTL 'MAP_PRIV_SECTION - Map private section'
1 ROUTINE MAP_PRIV_SECTION =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine attempts to open a user's private copy of TERMTABLE.EXE.
1 !       If successful, TERMTABLE.EXE is mapped into the user's virtual address
1 !       space as a temporary section.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = MAP_PRIV_SECTION ()
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       status returned by $OPEN
1 !       status returned by $CRMPSC
1 !       SS$_NORMAL
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    OWN
2
2       TABLE_NAME : VECTOR[25,BYTE]
2                    INITIAL(BYTE('TERM$TABLOC:TERMTABLE.EXE'));
2
2    LOCAL
2       REQ_ADDR : VECTOR [2] INITIAL (%X'200',%X'200'),
2                                       ! input address to $MGBLSC
2       ACTUAL_ADDR : VECTOR [2],       ! retd address from $MGBLSC
2       PRIVATE_FAB : $FAB (FOP = UFO, FAC=<GET>,SHR=<GET,UPI>),
2       OPEN_STATUS,
2       SECTION_STATUS;
2
2 !+
2 ! Initialize some fields.  This can't be done at link time because
2 ! it would require fixup vectors which would make the PSECT read/write
2 ! which would prevent sharing in our shared image.
2 !-
```

```
        PRIVATE_FAB[FAB$L_FNA]=TABLE_NAME;
        PRIVATE_FAB[FAB$B_FNS]=%ALLOCATION(TABLE_NAME);
    !+
    ! Look in TERM$TABLOC for private copy of TERMTABLE.EXE.
    !-
        IF (OPEN_STATUS = $OPEN (FAB = PRIVATE_FAB))
        THEN
            BEGIN
            !+
            ! Found a private termtable.  Map to it so we can determine
            ! if the requested terminal is defined there.
            !-
P           IF NOT (SECTION_STATUS = $CRMPSC (INADR = REQ_ADDR,
P                                            CHAN = .PRIVATE_FAB [FAB$L_STV],
P                                            FLAGS = SEC$M_EXPREG,
                                             RETADR = ACTUAL_ADDR))
            THEN
                RETURN (.SECTION_STATUS);

            PRIV_SECTION_ADDR = .ACTUAL_ADDR [0];
                                             ! save mapped addr
            PRIV_SECTION_ADDR_END = .ACTUAL_ADDR [1];
                                             ! also save end addr
            PRIV_SECTION_CHANNEL = .PRIVATE_FAB [FAB$L_STV];
                                             ! save channel for $DASSGN

            END
        ELSE
            RETURN (.OPEN_STATUS);

        RETURN (SS$_NORMAL);
        END;                         ! end of routine MAP_PRIV_SECTION


                                                      .PSECT  _SMG$DATA,NOEXE,  PIC,2

                                               00017  .BLKB   1
52 45 54 3A 43 4F 4C 42 41 54 24 4D 52 45 54   00018 TABLE_NAME:
                                                      .ASCII  \TERM$TABLOC:TERMTABLE.EXE\           ;
               45 58 45 2E 45 4C 42 41 54 4D   00027

                                                      .PSECT  _SMG$CODE,NOWRT,  SHR,  PIC,2

                                               003ED  .BLKB   3
                                         03    003F0 P.AAB:  .BYTE   3                            ;
                                         50    003F1  .BYTE   80                                   ;
                                       0000    003F2  .WORD   0                                    ;
                                   00020000    003F4  .LONG   131072                               ;
                                   00000000    003F8  .LONG   0                                    ;
                                   00000000    003FC  .LONG   0                                    ;
                                   00000000    00400  .LONG   0                                    ;
                                       0000    00404  .WORD   0                                    ;
                                         02    00406  .BYTE   2                                    ;
                                         42    00407  .BYTE   66                                   ;
                                   00000000    00408  .LONG   0                                    ;
```

```
                                00  0040C          .BYTE   0                                        ;
                                00  0040D          .BYTE   0                                        ;
                                00  0040E          .BYTE   0                                        ;
                                02  0040F          .BYTE   2                                        ;
                          00000000  00410          .LONG   0                                        ;
                          00000000  00414          .LONG   0                                        ;
                          00000000  00418          .LONG   0                                        ;
                          00000000  0041C          .LONG   0                                        ;
                          00000000  00420          .LONG   0                                        ;
                                00  00424          .BYTE   0                                        ;
                                00  00425          .BYTE   0                                        ;
                              0000  00426          .WORD   0                                        ;
                          00000000  00428          .LONG   0                                        ;
                              0000  0042C          .WORD   0                                        ;
                                00  0042E          .BYTE   0                                        ;
                                00  0042F          .BYTE   0                                        ;
                          00000000  00430          .LONG   0                                        ;
                          00000000  00434          .LONG   0                                        ;
                              0000  00438          .WORD   0                                        ;
                                00  0043A          .BYTE   0                                        ;
                                00  0043B          .BYTE   0                                        ;
                          00000000  0043C          .LONG   0                                        ;

                                                   .EXTRN  SYS$OPEN, SYS$CRMPSC

                              007C 00000 MAP_PRIV_SECTION:
                                                   .WORD   Save R2,R3,R4,R5,R6                      ; 1006
              56 00000000'   EF   9E 00002         MOVAB   TABLE_NAME, R6                           ;
              5E        A0   AE   9E 00009         MOVAB   -96(SP), SP                              ;
        58    AE      0200   8F   3C 0000D         MOVZWL  #512, REQ_ADDR                           ; 1042
        5C    AE      0200   8F   3C 00013         MOVZWL  #512, REQ_ADDR+4                         ;
  6E    91    AF      0050   8F   28 00019         MOVC3   #80, P.AAB, PRIVATE_FAB                  ; 1053
        2C    AE             66   9E 00020         MOVAB   TABLE_NAME, PRIVATE_FAB+44               ; 1062
        34    AE             19   90 00024         MOVB    #25, PRIVATE_FAB+52                      ; 1063
                             5E   DD 00028         PUSHL   SP                                       ; 1068
     00000000G    00         01   FB 0002A         CALLS   #1, SYS$OPEN                             ;
                  35         50   E9 00031         BLBC    OPEN_STATUS, 1$                          ;
                             7E   7C 00034         CLRQ    -(SP)                                    ; 1078
                             7E   7C 00036         CLRQ    -(SP)                                    ;
                        1C   AE   DD 00038         PUSHL   PRIVATE_FAB+12                           ;
                             7E   7C 0003B         CLRQ    -(SP)                                    ;
                             7E   D4 0003D         CLRL    -(SP)                                    ;
                  00020000   8F   DD 0003F         PUSHL   #131072                                  ;
                             7E   D4 00045         CLRL    -(SP)                                    ;
                        78   AE   9F 00047         PUSHAB  ACTUAL_ADDR                              ;
                        F8   AD   9F 0004A         PUSHAB  REQ_ADDR                                 ;
     00000000G    00         0C   FB 0004D         CALLS   #12, SYS$CRMPSC                          ;
                  12         50   E9 00054         BLBC    SECTION_STATUS, 1$                       ;
        EC        A6    50   AE   D0 00057         MOVL    ACTUAL_ADDR, PRIV_SECTION_ADDR           ; 1082
        F0        A6    54   AE   D0 0005C         MOVL    ACTUAL_ADDR+4, PRIV_SECTION_ADDR_END     ; 1084
        F4        A6    0C   AE   D0 00061         MOVL    PRIVATE_FAB+12, PRIV_SECTION_CHANNEL     ; 1086
                  50         01   D0 00066         MOVL    #1, R0                                   ; 1093
                             04 00069 1$:          RET                                              ; 1094

; Routine Size:  106 bytes,     Routine Base:  _SMG$CODE + 0440
```

```
; 1024          1095  1
```

```
%SBTTL 'MAP_GBL_SECTION - Map global section'
ROUTINE MAP_GBL_SECTION =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine maps the global section SMG$TERMTABLE into
!       the caller's virtual address space.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = MAP_GBL_SECTION ()
!
! FORMAL PARAMETERS:
!
!       NONE
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       status returned by $MGBLSC
!       SS$_NORMAL
!
! SIDE EFFECTS:
!
!       NONE
!--

    BEGIN
    LOCAL
        REQ_ADDR : VECTOR [2] INITIAL (%X'200',%X'200'),
                                        ! input address to $MGBLSC
        ACTUAL_ADDR : VECTOR [2],       ! retd address from $MGBLSC
        MAP_STATUS,                     ! status retd by $MGBLSC
        GBL_NAME_DESC : BLOCK [8,BYTE]; ! desc for global section name

    GBL_NAME_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
    GBL_NAME_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
    GBL_NAME_DESC [DSC$W_LENGTH] = %CHARCOUNT ('SMG$TERMTABLE');
    GBL_NAME_DESC [DSC$A_POINTER] = UPLIT ('SMG$TERMTABLE');

P   MAP_STATUS = $MGBLSC (INADR = REQ_ADDR,
P                         RETADR = ACTUAL_ADDR,
P                         FLAGS = SEC$M_SYSGBL + SEC$M_PERM + SEC$M_EXPREG,
                          GSDNAM = GBL_NAME_DESC);
    IF NOT .MAP_STATUS
    THEN
        RETURN .MAP_STATUS;

    GBL_SECTION_ADDR = .ACTUAL_ADDR [0];
```

```
; 1083          1153  2
; 1084          1154  2      RETURN (SS$_NORMAL);
; 1085          1155  1      END;                                      ! end of routine MAP_GBL_SECTION


                                                   004AA         .BLKB    2
00  00  45  4C  42  41  54  4D  52  45  54  24  47  4D  53  004AC P.AAC:  .ASCII   \SMG$TERMTABLE\<0><0><0>                    ;
                                               00  004BB                                                                    ;

                                                                 .EXTRN   SYS$MGBLSC

                                              0000 00000 MAP_GBL_SECTION:
                                                                 .WORD    Save nothing                                   ; 1097
                        5E             14  C2 00002              SUBL2    #20, SP                                        ;
                  0C    AE     0200    8F  3C 00005              MOVZWL   #512, REQ_ADDR                                 ; 1131
                  10    AE     0200    8F  3C 0000B              MOVZWL   #512, REQ_ADDR+4                               ;
                           010E000D    8F  DD 00011              PUSHL    #17694733                                      ; 1141
                  04    AE       D6    AF  9E 00017              MOVAB    P.AAC, GBL_NAME_DESC+4                         ; 1142
                                       7E  7C 0001C              CLRQ     -(SP)                                          ; 1147
                                 08    AE  9F 0001E              PUSHAB   GBL_NAME_DESC                                  ;
                           0002C000    8F  DD 00021              PUSHL    #180224                                        ;
                                       7E  D4 00027              CLRL     -(SP)                                          ;
                                 1C    AE  9F 00029              PUSHAB   ACTUAL_ADDR                                    ;
                                 28    AE  9F 0002C              PUSHAB   REQ_ADDR                                       ;
             00000000G  00             07  FB 0002F              CALLS    #7, SYS$MGBLSC                                 ;
                        0B             50  E9 00036              BLBC     MAP_STATUS, 1$                                 ; 1148
             00000000'  EF       08    AE  D0 00039              MOVL     ACTUAL_ADDR, GBL_SECTION_ADDR                  ; 1152
                        50             01  D0 00041              MOVL     #1, R0                                         ; 1154
                                           04 00044 1$:          RET                                                     ; 1155

; Routine Size:  69 bytes,    Routine Base:  _SMG$CODE + 04BC


; 1086          1156  1
```

```
1 %SBTTL 'SEARCH_FOR_TERM_DEF - Search for terminal definition'
1 ROUTINE SEARCH_FOR_TERM_DEF(
1                       TERM_TAB : REF BLOCK [,BYTE],
1                       TERM_NAME : REF BLOCK [,BYTE],
1                       RET_TERM_DEF      ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine searches a Termtable.exe for a particular
1 !       terminal definition.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       found.wl.v = SEARCH_FOR_TERM_DEF (TERM_TAB.rl.v,
1 !                                         TERM_NAME.rt.dx,
1 !                                         RET_TERM_DEF.wl.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TAB                address where TERMTABLE.EXE starts
1 !
1 !       TERM_NAME               name of the terminal for which we
1 !                               will search
1 !
1 !       RET_TERM_DEF            returned terminal definition address,
1 !                               if found
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       1 = terminal definition not found
1 !       0 = terminal definition was found
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2     LOCAL
2         FOUND : INITIAL (1),          ! flag to indicate end of search
2         BLOCK_NUMBER : BYTE,
2         INDEX_NAME_DESC : BLOCK [8,BYTE],
2         INDEX_NAME_COUNT : REF VECTOR [,BYTE],
2         INDEX_NAME_STRING : REF VECTOR [,BYTE];
2
2
2 !+
2 ! Each TERMTABLE.EXE contains an index of all the terminals defined.
```

**F

```
! Search the terminal index for the requested name.
!-

    INDEX_NAME_COUNT = .TERM_TAB + .TERM_TAB [TTB_L_INDEX_OFFSET];
                                        ! point to 1st count
                                        ! (address plus offset to get to
                                        !  start of index - count is 1st byte)
    INDEX_NAME_STRING = .INDEX_NAME_COUNT + 1;
                                        ! point to 1st name string in index

    IF .TERM_TAB [TTB_W_IDENT] NEQ 1
    THEN
        RETURN (SMG$_TABID_MIS);        ! Termtable Id Mismatch

    INDEX_NAME_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
    INDEX_NAME_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;

    WHILE .FOUND NEQ K_MATCH DO
        BEGIN
        !+
        ! Loop through all terminal names in the section
        ! index.  Exit if we find a match.
        !-
        INDEX_NAME_DESC [DSC$A_POINTER] = INDEX_NAME_STRING [0];
        INDEX_NAME_DESC [DSC$W_LENGTH] = .INDEX_NAME_COUNT [0];

        FOUND = STR$CASE_BLIND_COMPARE (.TERM_NAME, INDEX_NAME_DESC);

        IF .FOUND EQL K_MATCH
        THEN
            EXITLOOP
        ELSE
            BEGIN
            INDEX_NAME_COUNT = INDEX_NAME_STRING [0] + .INDEX_NAME_COUNT [0] + 1;
                                ! point past the previous string & block number
            IF .INDEX_NAME_COUNT [0] EQL 0
            THEN
                EXITLOOP;       ! return - no more terminals
            INDEX_NAME_STRING = .INDEX_NAME_COUNT + 1;
                                ! point to next name string
            END;

        END;                            ! end of WHILE loop

    !+
    ! If we found a match, use the block number stored with the index
    ! name to calculate the address of where the terminal definition
    ! starts.
    !-

    IF .FOUND EQL K_MATCH
    THEN
        BEGIN
        BLOCK_NUMBER = .(INDEX_NAME_STRING [0] + .INDEX_NAME_COUNT [0]);
                                        ! block # follows counted name string
        .RET_TERM_DEF = .TERM_TAB + ((.BLOCK_NUMBER - 1) * 512);
    END;                                ! end of search for requested name
```

```
; 1202           1271  2
; 1203           1272  2
; 1204           1273  2       RETURN (.FOUND);
; 1205           1274  1       END;                                        ! end of SEARCH_FOR_TERM_DEF



                                         003C 00000 SEARCH_FOR_TERM_DEF:
                                                              .WORD   Save R2,R3,R4,R5                              ; 1158
                         5E          08   C2 00002            SUBL2   #8, SP                                        :
                         50          01   D0 00005            MOVL    #1, FOUND                                     ; 1203
                         55    04    AC   D0 00008            MOVL    TERM_TAB, R5                                  ; 1217
               53        55    04    A5   C1 0000C            ADDL3   4(R5), R5, INDEX_NAME_COUNT                   :
                         52    01    A3   9E 00011            MOVAB   1(R3), INDEX_NAME_STRING                      ; 1221
                         01          65   B1 00015            CMPW    (R5), #1                                      ; 1224
                                     0B   13 00018            BEQL    1$                                            :
                         51 00000000G 00  9E 0001A            MOVAB   SMG$_TABID_MIS, R1                            ; 1226
                         50          51   D0 00021            MOVL    R1, R0                                        :
                                          04 00024            RET                                                   :
                    02   AE  010E    8F   B0 00025 1$:        MOVW    #270, INDEX_NAME_DESC+2                       ; 1228
                                     50   D5 0002B 2$:        TSTL    FOUND                                         ; 1231
                                     29   13 0002D            BEQL    3$                                            :
                    04   AE          52   D0 0002F            MOVL    INDEX_NAME_STRING, INDEX_NAME_DESC+4          ; 1237
                         54          63   9A 00033            MOVZBL  (INDEX_NAME_COUNT), R4                        ; 1238
                         6E          54   B0 00036            MOVW    R4, INDEX_NAME_DESC                           :
                                     5E   DD 00039            PUSHL   SP                                            ; 1240
                               08    AC   DD 0003B            PUSHL   TERM_NAME                                     :
               00000000G 00          02   FB 0003E            CALLS   #2, STR$CASE_BLIND_COMPARE                    :
                                     50   D5 00045            TSTL    FOUND                                         ; 1242
                                     0F   13 00047            BEQL    3$                                            :
                         53    01  A442   9E 00049            MOVAB   1(R4)[INDEX_NAME_STRING], INDEX_NAME_COUNT    ; 1247
                                     63   95 0004E            TSTB    (INDEX_NAME_COUNT)                            ; 1249
                                     06   13 00050            BEQL    3$                                            :
                         52    01    A3   9E 00052            MOVAB   1(R3), INDEX_NAME_STRING                      ; 1252
                                     D3   11 00056            BRB     2$                                            ; 1231
                                     50   D5 00058 3$:        TSTL    FOUND                                         ; 1264
                                     15   12 0005A            BNEQ    4$                                            :
                         51          63   9A 0005C            MOVZBL  (INDEX_NAME_COUNT), R1                        ; 1267
                         54        6142   90 0005F            MOVB    (R1)[INDEX_NAME_STRING], BLOCK_NUMBER         :
                         51          54   9A 00063            MOVZBL  BLOCK_NUMBER, R1                              ; 1269
               51        51          09   78 00066            ASHL    #9, R1, R1                                    :
                    0C   BC  FE00  C145   9E 0006A            MOVAB   -512(R1)[R5], @RET_TERM_DEF                   :
                                          04 00071 4$:        RET                                                   ; 1274

; Routine Size:  114 bytes,    Routine Base:  _SMG$CODE + 0501
```

```
%SBTTL 'SEARCH_FOR_TERM_DEF_BY_TYPE - search for term def given device type'
ROUTINE SEARCH_FOR_TERM_DEF_BY_TYPE (
                        TERM_TAB : REF BLOCK [,BYTE],
                        TERM_TYPE : SIGNED,
                        RET_TERM_DEF,
                        RET_TERM_NAME    ) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine searches a Termtable.exe for a particular
!       terminal definition.  It will look in all terminal definitions
!       for the given device type.
!
! CALLING SEQUENCE:
!
!       found.wl.v = SEARCH_FOR_TERM_DEF_BY_TYPE (TERM_TAB.rl.v,
!                                                  TERM_TYPE.rl.v,
!                                                  RET_TERM_DEF.wl.r,
!                                                  RET_TERM_NAME.wl.r)
!
! FORMAL PARAMETERS:
!
!       TERM_TAB                address where TERMTABLE.EXE starts
!
!       TERM_TYPE               type of the terminal for which we
!                               will search (signed longword value)
!
!       RET_TERM_DEF            returned terminal definition address,
!                               if found
!
!       RET_TERM_NAME           address of name associated with this device type
!                               (from the definition)
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       1 = terminal definition not found
!       0 = terminal definition was found
!
! SIDE EFFECTS:
!
!       NONE
!--

    BEGIN
    LOCAL
        FOUND : INITIAL (1),             ! flag to indicate end of search
        LOCAL_TYPE : SIGNED,             ! working terminal type
        CUR_TERM_DEF : REF VECTOR [,WORD], ! offset pointers are actually at
```

```
2                                       ! beginning of definition
2               INDEX_NAME_COUNT : REF VECTOR [,BYTE],
2               INDEX_NAME_STRING : REF VECTOR [,BYTE];
2
2
2       !+
2       ! Each TERMTABLE.EXE contains an index of all the terminals defined.
2       ! Look at the definition of every terminal in the index, until we
2       ! find one with the requested device type.
2       !-
2
2           INDEX_NAME_COUNT = .TERM_TAB + .TERM_TAB [TTB_L_INDEX_OFFSET];
2                                           ! point to 1st count
2                                           ! (address plus offset to get to
2                                           !  start of index - count is 1st byte)
2           INDEX_NAME_STRING = .INDEX_NAME_COUNT + 1;
2                                           ! point to 1st name string in index
2
2           IF .TERM_TAB [TTB_W_IDENT] NEQ 1
2           THEN
2               RETURN (SMG$_TABID_MIS);        ! Termtable Id Mismatch
2
2       !+
2       ! If this definition is in a private section, the type number was biased.
2       ! Since the unbiased number is what is actually in the file, remove the
2       ! bias before doing any comparisons.
2       !-
2
2           LOCAL_TYPE = .TERM_TYPE;
2           $UNBIAS_TYPE_NUMBER (.TERM_TAB, LOCAL_TYPE);
2
2           WHILE .FOUND NEQ K_MATCH DO
3               BEGIN
3               LOCAL
3                   BLOCK_NUMBER : BYTE,
3                   DEF_DEVICE : REF VECTOR [,BYTE,SIGNED];
3                                           ! points to term number cap
3               !+
3               ! Loop through all terminal names in the section
3               ! index.  Exit if we find a match.
3               !-
3               BLOCK_NUMBER = .(INDEX_NAME_STRING [0] + .INDEX_NAME_COUNT [0]);
3                                           ! block # follows counted name string
3               CUR_TERM_DEF = .TERM_TAB + ((.BLOCK_NUMBER - 1) * 512);
3                                           ! get def for this index entry
3               DEF_DEVICE = .CUR_TERM_DEF [SMG$K_VMS_TERMINAL_NUMBER] +
3                            .CUR_TERM_DEF;
3                                           ! point to device type in definition
3               !+
3               ! Device type is stored as a signed longword.  Skip past the
3               ! size and type bytes to the actual data.
3               !-
3               IF .DEF_DEVICE [2] EQL .LOCAL_TYPE
3               THEN
4                   BEGIN
4                   FOUND = K_MATCH;
4                   EXITLOOP
```

```
4                 END
3             ELSE
4                 BEGIN
4                 INDEX_NAME_COUNT = INDEX_NAME_STRING [0] + .INDEX_NAME_COUNT [0] +
4                                         1;
4                                 ! point past the previous string & block number
4                 IF .INDEX_NAME_COUNT [0] EQL 0
4                 THEN
4                     EXITLOOP;           ! return - no more terminals
4                 INDEX_NAME_STRING = .INDEX_NAME_COUNT + 1;
4                                 ! point to next name string
3                 END;
3
2             END;                        ! end of WHILE loop
2
2     !+
2     ! If we found a match, set the return name and return definition address.
2     !-
2
2         IF .FOUND EQL K_MATCH
2         THEN
3             BEGIN
3             .RET_TERM_DEF = .CUR_TERM_DEF;
3             .RET_TERM_NAME = .CUR_TERM_DEF + .CUR_TERM_DEF [SMG$K_NAME];
3                                 ! address of counted string
2             END;
2
2         RETURN (.FOUND);
1         END;                            ! end of SEARCH_FOR_TERM_DEF_BY_TYPE



                                  01FC 00000 SEARCH_FOR_TERM_DEF_BY_TYPE:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8                ; 1276
              58 00000000'  EF  9E 00002                MOVAB   PRIV_SECTION_ADDR, R8
              57            01  D0 00009                MOVL    #1, FOUND                                ; 1327
              55        04  AC  D0 0000C                MOVL    TERM_TAB, R5                             ; 1343
     51       55        04  A5  C1 00010                ADDL3   4(R5), R5, INDEX_NAME_COUNT
              50        01  A1  9E 00015                MOVAB   1(R1), INDEX_NAME_STRING                 ; 1347
              01            65  B1 00019                CMPW    (R5), #1                                 ; 1350
                            0B  13 0001C                BEQL    1$                                       ;
              52 00000000G  00  9E 0001E                MOVAB   SMG$_TABID_MIS, R2                       ; 1352
              50            52  D0 00025                MOVL    R2, R0                                   ;
                                04 00028                RET                                              ;
              56        08  AC  D0 00029 1$:            MOVL    TERM_TYPE, LOCAL_TYPE                    ; 1360
                            68  D5 0002D                TSTL    PRIV_SECTION_ADDR                        ; 1361
                            0D  13 0002F                BEQL    2$                                       ;
              68            55  D1 00031                CMPL    R5, PRIV_SECTION_ADDR                    ;
                            08  19 00034                BLSS    2$                                       ;
                            56  D5 00036                TSTL    LOCAL_TYPE                               ;
                            04  18 00038                BGEQ    2$                                       ;
              56        41  A6  9E 0003A                MOVAB   65(R6), LOCAL_TYPE                       ;
                            57  D5 0003E 2$:            TSTL    FOUND                                    ; 1363
                            37  13 00040                BEQL    4$                                       ;
              54            61  9A 00042                MOVZBL  (INDEX_NAME_COUNT), R4                   ; 1373
```

```
                      52        6440  90 00045        MOVB    (R4)[INDEX_NAME_STRING], BLOCK_NUMBER          ;
                      52          52  9A 00049        MOVZBL  BLOCK_NUMBER, R2                               ; 1375
            52        52          09  78 0004C        ASHL    #9, R2, R2                                     ;
                      53   FE00 C245  9E 00050        MOVAB   -512(R2)[R5], CUR_TERM_DEF                     ;
                      52   01C6   C3  3C 00056        MOVZWL  454(CUR_TERM_DEF), DEF_DEVICE                  ; 1378
                      52          53  C0 0005B        ADDL2   CUR_TERM_DEF, DEF_DEVICE                       ;
  56      02  A2      08          00  EC 0005E        CMPV    #0, #8, 2(DEF_DEVICE), LOCAL_TYPE              ; 1384
                                  04  12 00064        BNEQ    3$                                             ;
                                  57  D4 00066        CLRL    FOUND                                          ; 1387
                                  0F  11 00068        BRB     4$                                             ; 1386
                      51     01 A440  9E 0006A 3$:    MOVAB   1(R4)[INDEX_NAME_STRING], INDEX_NAME_COUNT     ; 1392
                                  61  95 0006F        TSTB    (INDEX_NAME_COUNT)                             ; 1395
                                  06  13 00071        BEQL    4$                                             ;
                      50     01   A1  9E 00073        MOVAB   1(R1), INDEX_NAME_STRING                       ; 1398
                                  C5  11 00077        BRB     2$                                             ; 1363
                                  57  D5 00079 4$:    TSTL    FOUND                                          ; 1408
                                  0E  12 0007B        BNEQ    5$                                             ;
               0C     BC          53  D0 0007D        MOVL    CUR_TERM_DEF, @RET_TERM_DEF                    ; 1411
                      50   0456   C3  3C 00081        MOVZWL  1110(CUR_TERM_DEF), R0                         ; 1412
      10   BC         53          50  C1 00086        ADDL3   R0, CUR_TERM_DEF, @RET_TERM_NAME               ;
                      50          57  D0 0008B 5$:    MOVL    FOUND, R0                                      ; 1416
                                      04 0008E        RET                                                    ; 1417
```

; Routine Size: 143 bytes, Routine Base: _SMG$CODE + 0573

```
1 %SBTTL 'DECODE_ARITH_STRING - Decode arithmetic data'
1 ROUTINE DECODE_ARITH_STRING (ARITH_STRING : REF VECTOR [,BYTE],
1                              USER_ARGS : REF VECTOR,
1                              DECODED_STRING : REF VECTOR [,BYTE],
1                              DECODED_LENGTH) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Arithmetic strings are stored as encoded data bytes.  Codes
1 !       are used to indicate where an operand is, what operator to apply,
1 !       etc.  The general format is number, op, number, op, number, ... etc.
1 !
1 !       In the midst of binary information, there may be text that must
1 !       be included in the final string.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       status.rl.v = DECODE_ARITH_STRING (ARITH_STRING.rl.r,
1 !                                          USER_ARGS.rl.r,
1 !                                          DECODED_STRING.wl.r,
1 !                                          DECODED_LENGTH.wl.r)
1 !
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       ARITH_STRING.rl.r               Address of string to decode
1 !
1 !       USER_ARGS.rl.r                  Address of a vector containing
1 !                                       args provided by caller
1 !
1 !       DECODED_STRING.wl.r             Address where decoded string
1 !                                       should be placed
1 !
1 !       DECODED_LENGTH.wl.r             Length of final string
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SUCCESS         String was decoded
1 !       FAILURE         String could not be decoded
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
2     BEGIN
2     LOCAL
2         ACCUMULATOR : INITIAL (0),      ! place to hold current result
2         INDEX,                          ! index into arithmetic string
```

```
2          CURRENT_BYTE : INITIAL (0),       ! pointer into return string
2          ARG_COUNT : INITIAL (0),          ! number of user args that have
2                                            ! been used
2          ARITH_CAP_SIZE,                   ! length of entire capability string
2          SEEN : INITIAL (0);               ! flag to indicate 1st operand found
2
2   !+
2   ! Initialize return length.
2   !-
2
2      .DECODED_LENGTH = 0;
2
2   !+
2   ! Start with the 3rd byte of the string (skipping over the size and
2   ! type of the entire string).  This byte should be the data type of
2   ! the first item to process.  It could be one of the following:
2   !       positive number          text
2   !       k_operand                longword operand
2   !       k_<operator>             add, subtract, multiply, divide
2   !       k_store                  calculation complete, store result
2   !-
2
2      ARITH_CAP_SIZE = .ARITH_STRING [0] XOR %X'FFFFFF00';
2                                            ! sign extend byte
2      ARITH_CAP_SIZE = ABS (.ARITH_CAP_SIZE);
2                                            ! get length (which is stored negative)
2
2      INDEX = 2;                            ! start with 3rd byte
2
2      WHILE .INDEX LSS .ARITH_CAP_SIZE DO
3          BEGIN
3          LOCAL
3              DATA_TYPE : BYTE,             ! code for kind of data to follow
3              DATA_SIZE : BYTE,             ! relevant only for string data
3              TEMP : REF VECTOR;            ! addr of operand awaiting operator
3
3          DATA_TYPE = .ARITH_STRING [.INDEX];
3
3          SELECTONE .DATA_TYPE OF
3          SET
3
3              [.K_OPERAND]:                 ! longword operand
4                  BEGIN
4                  IF NOT .SEEN
4                  THEN
5                      BEGIN                 ! 1st operand seen goes in accum
5                      CH$MOVE (4, ARITH_STRING [.INDEX + 2], ACCUMULATOR);
5                      SEEN = 1;
5                      END
4                  ELSE
4                      TEMP = ARITH_STRING [.INDEX + 2];
4                                            ! remember location of operand
4                  INDEX = .INDEX + 6;       ! get next code
3                  END;
3
3              [.K_SUBSTITUTE]:              ! substitute argument from user
4                  BEGIN
```

```
            LOCAL
                ARG_NUMBER;
            !+
            ! Number following substitution symbol will tell us
            ! which argument from the vector to use.
            !-
            ARG_NUMBER = .ARITH_STRING [.INDEX+1];
            IF NOT .SEEN
            THEN
                BEGIN                   ! 1st operand found goes into accum
                ACCUMULATOR = .USER_ARGS [.ARG_NUMBER-1];
                SEEN = 1;
                END
            ELSE
                TEMP = USER_ARGS [.ARG_NUMBER-1];
                                        ! remember location of operand
            INDEX = .INDEX + 2;         ! next code
            END;

        [.K_ADD]:
            BEGIN                       ! add temp to accumulator
            ACCUMULATOR = .TEMP [0] + .ACCUMULATOR;
            TEMP = 0;                   ! don't use this operand again
            INDEX = .INDEX + 1;         ! next code
            END;

        [.K_SUBTRACT]:
            BEGIN                       ! subtract temp from accumulator
            ACCUMULATOR = .ACCUMULATOR - .TEMP [0];
            TEMP = 0;                   ! don't reuse this operand
            INDEX = .INDEX + 1;         ! next code
            END;

        [.K_MULTIPLY]:
            BEGIN                       ! multiply temp by accumulator
            ACCUMULATOR = .ACCUMULATOR * .TEMP [0];
            TEMP = 0;                   ! don't reuse this operand
            INDEX = .INDEX + 1;         ! next code
            END;

        [.K_DIVIDE]:
            BEGIN                       ! divide accumulator by temp
            ACCUMULATOR = .ACCUMULATOR / .TEMP [0];
            TEMP = 0;                   ! don't reuse this operand
            INDEX = .INDEX + 1;         ! next code
            END;

        [.K_STORE]:
            BEGIN                       ! store a calculation
            DECODED_STRING [.CURRENT_BYTE] = .ACCUMULATOR <0,8>;
                                        ! store only as a byte
            CURRENT_BYTE = .CURRENT_BYTE + 1;
            ACCUMULATOR = 0;
            SEEN = 0;
            INDEX = .INDEX + 1;         ! next code
            END;
```

```
; 1522      1589 3            [OTHERWISE]:                      ! plain vanilla text
; 1523      1590 4                BEGIN
; 1524      1591 4                CH$MOVE (.ARITH_STRING [.INDEX],        ! size of text
; 1525      1592 4                         ARITH_STRING [.INDEX + 2],     ! skip over type byte
; 1526      1593 4                         DECODED_STRING [.CURRENT_BYTE]);
; 1527      1594 4                CURRENT_BYTE = .CURRENT_BYTE + .ARITH_STRING [.INDEX];
; 1528      1595 4                INDEX = .INDEX + .ARITH_STRING [.INDEX] + 2;
; 1529      1596 4                                                  ! move past string, count & type bytes
; 1530      1597 3                END;
; 1531      1598 3        TES;
; 1532      1599 3
; 1533      1600 2        END;                                      ! end of WHILE LOOP
; 1534      1601 2
; 1535      1602 2    .DECODED_LENGTH = .CURRENT_BYTE;              ! return length of string
; 1536      1603 2
; 1537      1604 2    RETURN (SS$_NORMAL);
; 1538      1605 2
; 1539      1606 1    END;                                          ! end of routine DECODE_ARITH_STRING
; INFO#250              L1:1553
; Referenced LOCAL symbol TEMP is probably not initialized



                                  0FFC 00000 DECODE_ARITH_STRING:
                                                     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1419
                             7E   D4 00002           CLRL    ACCUMULATOR                                   ; 1471
                             58   D4 00004           CLRL    CURRENT_BYTE                                  ;
                             50   D4 00006           CLRL    ARG_COUNT                                     ;
                             5B   D4 00008           CLRL    SEEN                                          ;
                      10     BC   D4 0000A           CLRL    @DECODED_LENGTH                               ; 1485
           56         04     AC   D0 0000D           MOVL    ARITH_STRING, R6                              ; 1497
           5A                66   9A 00011           MOVZBL  (R6), ARITH_CAP_SIZE                          ;
           5A FFFFFF00       8F   CC 00014           XORL2   #-256, ARITH_CAP_SIZE                         ;
                             03   18 0001B           BGEQ    1$                                            ; 1499
           5A                5A   CE 0001D           MNEGL   ARITH_CAP_SIZE, ARITH_CAP_SIZE                ;
           57                02   D0 00020 1$:       MOVL    #2, INDEX                                     ; 1502
           5A                57   D1 00023 2$:       CMPL    INDEX, ARITH_CAP_SIZE                         ; 1504
                             03   19 00026           BLSS    3$                                            ;
                           00B5   31 00028           BRW     19$                                           ;
           59              6746   9A 0002B 3$:       MOVZBL  (INDEX)[R6], R9                               ; 1511
           52                59   90 0002F           MOVB    R9, DATA_TYPE                                 ;
  00000000' EF               52   91 00032           CMPB    DATA_TYPE, K_OPERAND                          ; 1516
                             19   12 00039           BNEQ    7$                                            ;
           0C                5B   E8 0003B           BLBS    SEEN, 4$                                      ; 1518
                        02 A746   9F 0003E           PUSHAB  2(INDEX)[R6]                                  ; 1521
           6E                9E   D0 00042           MOVL    @(SP)+, ACCUMULATOR                           ;
           5B                01   D0 00045           MOVL    #1, SEEN                                      ; 1522
                             05   11 00048           BRB     5$                                            ; 1518
           51           02 A746   9E 0004A 4$:       MOVAB   2(INDEX)[R6], TEMP                            ; 1525
           57                06   C0 0004F 5$:       ADDL2   #6, INDEX                                     ; 1527
                             CF   11 00052 6$:       BRB     2$                                            ; 1513
  00000000' EF               52   91 00054 7$:       CMPB    DATA_TYPE, K_SUBSTITUTE                       ; 1530
                             1F   12 0005B           BNEQ    11$                                           ;
           50           01 A746   9A 0005D           MOVZBL  1(INDEX)[R6], ARG_NUMBER                      ; 1538
           50           08 BC40   DE 00062           MOVAL   @USER_ARGS[ARG_NUMBER], R0                    ; 1542
```

```
                      09            5B E8 00067        BLBS    SEEN, 8$                                   ; 1539
                      6E       FC   A0 D0 0006A        MOVL    -4(R0), ACCUMULATOR                        ; 1542
                      5B            01 D0 0006E        MOVL    #1, SEEN                                   ; 1543
                                    04 11 00071        BRB     9$                                         ; 1539
                      51       FC   A0 9E 00073 8$:    MOVAB   -4(R0), TEMP                               ; 1546
                      57            02 C0 00077 9$:    ADDL2   #2, INDEX                                  ; 1548
                                    A7 11 0007A 10$:   BRB     2$                                         ; 1513
             00000000'  EF          52 91 0007C 11$:   CMPB    DATA_TYPE, K_ADD                           ; 1551
                                    05 12 00083        BNEQ    12$                                        ;
                      6E            61 C0 00085        ADDL2   (TEMP), ACCUMULATOR                        ; 1553
                                    28 11 00088        BRB     15$                                        ; 1554
             00000000'  EF          52 91 0008A 12$:   CMPB    DATA_TYPE, K_SUBTRACT                      ; 1558
                                    05 12 00091        BNEQ    13$                                        ;
                      6E            61 C2 00093        SUBL2   (TEMP), ACCUMULATOR                        ; 1560
                                    1A 11 00096        BRB     15$                                        ; 1561
             00000000'  EF          52 91 00098 13$:   CMPB    DATA_TYPE, K_MULTIPLY                      ; 1565
                                    05 12 0009F        BNEQ    14$                                        ;
                      6E            61 C4 000A1        MULL2   (TEMP), ACCUMULATOR                        ; 1567
                                    0C 11 000A4        BRB     15$                                        ; 1568
             00000000'  EF          52 91 000A6 14$:   CMPB    DATA_TYPE, K_DIVIDE                        ; 1572
                                    07 12 000AD        BNEQ    16$                                        ;
                      6E            61 C6 000AF        DIVL2   (TEMP), ACCUMULATOR                        ; 1574
                                    51 D4 000B2 15$:   CLRL    TEMP                                       ; 1575
                                    14 11 000B4        BRB     17$                                        ; 1576
             00000000'  EF          52 91 000B6 16$:   CMPB    DATA_TYPE, K_STORE                         ; 1579
                                    0F 12 000BD        BNEQ    18$                                        ;
                 0C BC48            6E 90 000BF        MOVB    ACCUMULATOR, @DECODED_STRING[CURRENT_BYTE] ; 1581
                                    58 D6 000C4        INCL    CURRENT_BYTE                               ; 1583
                                    6E D4 000C6        CLRL    ACCUMULATOR                                ; 1584
                                    5B D4 000C8        CLRL    SEEN                                       ; 1585
                                    57 D6 000CA 17$:   INCL    INDEX                                      ; 1586
                                    84 11 000CC        BRB     6$                                         ; 1513
      0C BC48         02 A746       59 28 000CE 18$:   MOVC3   R9, 2(INDEX)[R6], @DECODED_STRING-         ; 1593
                                                               [CURRENT_BYTE]                             ;
                      58            59 C0 000D6        ADDL2   R9, CURRENT_BYTE                           ; 1594
                      57   02 A947  9E 000D9           MOVAB   2(R9)[INDEX], INDEX                        ; 1595
                                    9A 11 000DE        BRB     10$                                        ; 1504
                      10  BC        58 D0 000E0 19$:   MOVL    CURRENT_BYTE, @DECODED_LENGTH              ; 1602
                      50            01 D0 000E4        MOVL    #1, R0                                     ; 1604
                                       04 000E7        RET                                                ; 1606

; Routine Size:  232 bytes,    Routine Base:  _SMG$CODE + 0602
```

```
; 1541          1607  1     END                                   ! end of module SMG$INTERFACE_TERM_TABLE
; 1542          1608  0     ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;    _SMG$DATA                     49  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;    _SMG$CODE                   1770  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                             Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;    _$255$DUA28:[SYSLIB]STARLET.L32;1     9776       46         0          581        00:01.0
;    _$255$DUA28:[SMGRTL.OBJ]RTLLIB.L32;1    36        0         0            8        00:00.1
;    _$255$DUA28:[SMGRTL.OBJ]SMGLIB.L32;1   469        1         0           38        00:00.4
;    _$255$DUA28:[SMGRTL.OBJ]SMGTPALIB.L32;1
;                                            41       15        36           10        00:00.1



; Information:   1
; Warnings:      0
; Errors:        0



;                                   COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:SMGUSRTRM/OBJ=OBJ$:SMGUSRTRM MSRC$:SMGUSRTRM/UPDATE=(ENH$:SMGUSRTRM
;       )

; Size:          1627 code + 192 data bytes
; Run Time:         00:33.8
; Elapsed Time:     01:40.2
; Lines/CPU Min:     2858
; Lexemes/CPU-Min: 18090
; Memory Used:  173 pages
; Compilation Complete
```

**F

0362 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COM REQ

SMGUSRTRM LIS

SORLIB REQ

SORTSHR MAP

DEFS0 REQ

DKS REQ

SORT32

SRTSPC REQ

CDDTYPE R32

SORTMERGE MAP

SMGVECTOR LIS

CHKPNT REQ

CDDMAC R32

SFKEYWRD REQ

RECSYM R32

SRTTRN MAR

OPCODES REQ