M 7

```
   CCCCCCCCCCCC     OOOOOOOOO     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC     OOOOOOOOO     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC     OOOOOOOOO     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
CCC              OOO         OOO  BBB         BBB  RRR         RRR        TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR         RRR        TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR         RRR        TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR         RRR        TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR         RRR        TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR         RRR        TTT        LLL
CCC              OOO         OOO  BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
CCC              OOO         OOO  BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
CCC              OOO         OOO  BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR   RRR              TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR   RRR              TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR   RRR              TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR      RRR           TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR      RRR           TTT        LLL
CCC              OOO         OOO  BBB         BBB  RRR      RRR           TTT        LLL
   CCCCCCCCCCCC     OOOOOOOOO     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC     OOOOOOOOO     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC     OOOOOOOOO     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

I 10

```
**FILE**ID**COBESCGEN
```

```
  CCCCCCCC    000000    BBBBBBBB    EEEEEEEEEE    SSSSSSSS    CCCCCCCC    GGGGGGGG  EEEEEEEEEE  NN      NN
  CCCCCCCC    000000    BBBBBBBB    EEEEEEEEEE    SSSSSSSS    CCCCCCCC    GGGGGGGG  EEEEEEEEEE  NN      NN
CC          00      00  BB      BB  EE          SS          CC          GG          EE          NN      NN
CC          00      00  BB      BB  EE          SS          CC          GG          EE          NN      NN
CC          00      00  BB      BB  EE          SS          CC          GG          EE          NNNN    NN
CC          00      00  BB      BB  EE          SS          CC          GG          EE          NNNN    NN
CC          00      00  BBBBBBBB    EEEEEEEE      SSSSSS    CC          GG          EEEEEEEE    NN  NN  NN
CC          00      00  BBBBBBBB    EEEEEEEE      SSSSSS    CC          GG          EEEEEEEE    NN  NN  NN
CC          00      00  BB      BB  EE                  SS  CC          GG  GGGGGG  EE          NN    NNNN
CC          00      00  BB      BB  EE                  SS  CC          GG  GGGGGG  EE          NN    NNNN
CC          00      00  BB      BB  EE                  SS  CC          GG      GG  EE          NN      NN      ....
CC          00      00  BB      BB  EE                  SS  CC          GG      GG  EE          NN      NN      ....
  CCCCCCCC    000000    BBBBBBBB    EEEEEEEEEE  SSSSSSSS      CCCCCCCC    GGGGGG    EEEEEEEEEE  NN      NN      ....
  CCCCCCCC    000000    BBBBBBBB    EEEEEEEEEE  SSSSSSSS      CCCCCCCC    GGGGGG    EEEEEEEEEE  NN      NN      ....

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 'COB$$ESCAPE_GENERATOR - Escape sequence generator for screen mgmt'
0 MODULE COB$$ESCAPE_GENERATOR (
0               IDENT = '1-003' ! File: COBESCGEN.B32 Edit: STAN1003
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:     General Utility Library
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains routines which return a device-specific
1 !       escape sequence to perform a specified function.
1 !
1 !       These are low level routines; the burden of validity checking
1 !       is on the caller.  For example, buffers are allocated by the caller,
1 !       and these routines do not check for overflowing the buffers bounds.
1 !       If the device is not a video terminal, no escape sequence will be
1 !       generated, and the routine will return with a success status.
1 !
1 ! ENVIRONMENT:  User mode, Shared library routines.
1 !
1 ! AUTHOR: P. Levesque, CREATION DATE: 7-Mar-1983
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  PLL 7-Mar-1983
1 ! 1-002 - Add COB$$SET_ATTRIBUTES_ONLY.
1 !         Fix call to COB$$SET_CURSOR_ABS_R4 in COB$$SET_CURSOR_REL.
1 !         Fix to COB$$SET_CURSOR_REL.  If we are at the 1st column and the
1 !         previous character was a <CR>, then the terminal driver may give
1 !         us a 'free' <LF> on our next operation if it is a read.  To avoid
1 !         the problem, just make sure <CR> is not the last thing in the
```

```
;   58          0058  1 !               output buffer.
;   59          0059  1 !               Rename module from SMG$$ESCAPE_GENERATOR to COB$$ESCAPE_GENERATOR.
;   60          0060  1 !                                                             LGB 20-FEB-1984
;   61          0061  1 ! 1-003 - Removed informational errors. STAN 24-Jul-1984.
;   62          0062  1 !--
;   63          0063  1
```

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

!
! LINKAGES:
!
!       NONE
!
! INCLUDE FILES:
!
REQUIRE 'RTLIN:COBPROLOG';                      ! Defines psects, macros, &
                                                ! terminal defs
REQUIRE 'RTLIN:COBLNK';                         ! Linkages
!
! TABLE OF CONTENTS:
!

FORWARD ROUTINE

    COB$$DOWN_SCROLL_R2 : COB$$ESC_R2_LNK, ! Creat downscroll sequence
    COB$$ERASE_LINE_R2 : COB$$ESC_R2_LNK,  ! Create erase line sequence
    COB$$ERASE_PAGE_R2 : COB$$ESC_R2_LNK,  ! Create erase page sequence
    COB$$ERASE_WHOLE_LINE_R2 : COB$$ESC_R2_LNK, ! Create erase whole line sequence
    COB$$ERASE_WHOLE_PAGE_R2 : COB$$ESC_R2_LNK, ! Create erase whole page sequence
    COB$$SET_ATTRIBUTES,                   ! Create set attributes sequences w text
    COB$$SET_ATTRIBUTES_ONLY,              ! Create set attributes sequences w no text
    COB$$SET_CURSOR_ABS_R4 : COB$$ESC_R4_LNK,! Create absolute set cursor sequence
    COB$$SET_CURSOR_REL,                   ! Create relative set cursor sequence
    COB$$SETUP_TERM_TYPE,                  ! Setup terminal type for COB$$ calls
    COB$$UP_SCROLL_R2 : COB$$ESC_R2_LNK;   ! Create upscroll sequence

!
! MACROS:
!

!
! EQUATED SYMBOLS:
!

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

! OWN STORAGE:
!
!       NONE

!
! EXTERNAL REFERENCES:
!

```

```
;  122          1711  1 EXTERNAL ROUTINE
;  123          1712  1
;  124          1713  1     LIB$FREE_EF,                ! free event flag number
;  125          1714  1     LIB$GET_EF;                 ! get event flag number
;  126          1715  1
;  127          1716  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$DOWN_SCROLL_R2 - Create downscroll sequence'
1 GLOBAL ROUTINE COB$$DOWN_SCROLL_R2 (
1                         TERM_TYPE,
1                         BUFFER,
1                         CUR_SIZE
1                     ) : COB$$ESC_R2_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence for down scroll
1 !       and appends the string to a given output buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$DOWN_SCROLL_R2 (TERM_TYPE.rl.v, BUFFER.mt.r,
1 !                                               CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     LOCAL
2         FREE_ADDR;
2
2     BIND
2         VT05_DOWN = UPLIT (BYTE (CR, VT05_CUP, NULL)),
2         VT52_DOWN = UPLIT (BYTE (ESC, VT52_DWN)),
2         VT100_DOWN = UPLIT (BYTE (ESC, VT100_DWN));
2
2     FREE_ADDR = .BUFFER + ..CUR_SIZE;
2
2     CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
2     SET
2         [VT05]:
3             BEGIN
3             CH$MOVE (3, VT05_DOWN, .FREE_ADDR);
3             .CUR_SIZE = ..CUR_SIZE + 3;
```

```
;  186          1774  2              END;
;  187          1775  2
;  188          1776  2          [VT52]:
;  189          1777  3              BEGIN
;  190          1778  3              CH$MOVE (2, VT52_DOWN, .FREE_ADDR);
;  191          1779  3              .CUR_SIZE = ..CUR_SIZE + 2;
;  192          1780  2              END;
;  193          1781  2
;  194          1782  2          [VT100]:
;  195          1783  3              BEGIN
;  196          1784  3              CH$MOVE (2, VT100_DOWN, .FREE_ADDR);
;  197          1785  3              .CUR_SIZE = ..CUR_SIZE + 2;
;  198          1786  2              END;
;  199          1787  2
;  200          1788  2          [HARDCOPY, UNKNOWN, VTFOREIGN]:
;  201          1789  2              ;
;  202          1790  2
;  203          1791  2          [INRANGE, OUTRANGE]:
;  204          1792  2              RETURN 0;                   ! should never get here
;  205          1793  2
;  206          1794  2          TES;
;  207          1795  2
;  208          1796  2      RETURN (SS$_NORMAL);
;  209          1797  2
;  210          1798  1      END;


                                                             .TITLE  COB$$ESCAPE_GENERATOR COB$$ESCAPE_GENERATOR - E
;                                                                                          scape sequence generat
                                                             .IDENT  \1-003\

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

                                     00  1A  0D  00000 P.AAA:  .BYTE   13, 26, 0                                     ;
                                                 00003         .BLKB   1
                                         49  1B  00004 P.AAB:  .BYTE   27, 73                                        ;
                                                 00006         .BLKB   2
                                         4D  1B  00008 P.AAC:  .BYTE   27, 77                                        ;

                                                       VT05_DOWN=          P.AAA
                                                       VT52_DOWN=          P.AAB
                                                       VT100_DOWN=         P.AAC
                                                             .EXTRN  LIB$FREE_EF, LIB$GET_EF

                          51             62  C0 00000 COB$$DOWN_SCROLL_R2::
                                                             ADDL2   @CUR_SIZE), FREE_ADDR                        ; 1766
              05          00             50  CF 00003        CASEL   TERM_TYPE, #0, #5                            ; 1768
    001F    0019        000E           0026     00007 1$:    .WORD   6$-1$,-                                      ;
                        0026           0026     0000F                2$-1$,-                                      ;
                                                                     3$-1$,-                                      ;
                                                                     4$-1$,-                                      ;
                                                                     6$-1$,-                                      ;
                                                                     6$-1$                                        ;
                                         1C  11 00013        BRB     7$                                           ; 1792
      61          18          00     DE  AF  F0 00015 2$:    INSV    VT05_DOWN, #0, #24, (FREE_ADDR)              ; 1772
                              62         03  C0 0001B        ADDL2   #3, @CUR_SIZE)                               ; 1773
                                         0D  11 0001E        BRB     6$                                           ; 1768
```

```
                                   61      D7   AF  B0 00020 3$:      MOVW    VT52_DOWN, (FREE_ADDR)                  ; 1778
                                                04  11 00024          BRB     5$                                      ; 1779
                                   61      D5   AF  B0 00026 4$:      MOVW    VT100_DOWN, (FREE_ADDR)                 ; 1784
                                   62           02  C0 0002A 5$:      ADDL2   #2, (CUR_SIZE)                          ; 1785
                                   50           01  D0 0002D 6$:      MOVL    #1, R0                                  ; 1796
                                                    05 00030          RSB                                             ;
                                                50  D4 00031 7$:      CLRL    R0                                      ; 1798
                                                    05 00033          RSB                                             ;

; Routine Size:  52 bytes,    Routine Base:  _COB$CODE + 000A


;  211          1799  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$ERASE_LINE_R2 - Create erase line sequence'
1 GLOBAL ROUTINE COB$$ERASE_LINE_R2 (
1                       TERM_TYPE,
1                       BUFFER,
1                       CUR_SIZE
1                   ) : COB$$ESC_R2_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence for erasing a
1 !       line from the current cursor position.  The string is
1 !       appended to the given output buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$ERASE_LINE_R2 (TERM_TYPE.rl.v,
1 !                                   BUFFER.mt.r, CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !                                updated to reflect erase seq added
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    LOCAL
2       FREE_ADDR;                      ! addr of next free byte in buffer
2
2    BIND
2       VT05_LINE = UPLIT (BYTE (VT05_EOL, NULL, NULL)),
2       VT52_LINE = UPLIT (BYTE (ESC, VT52_EOL)),
2       VT100_LINE = UPLIT (BYTE (ESC, LB, VT100_EO' ));
2
2    FREE_ADDR = .BUFFER + ..CUR_SIZE;
2
2    CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
2    SET
2       [VT05]:
3           BEGIN
```

```
;  270          1857  3              CH$MOVE (3, VT05_LINE, .FREE_ADDR);
;  271          1858  3              .CUR_SIZE = ..CUR_SIZE + 3;
;  272          1859  2              END;
;  273          1860  2
;  274          1861  2          [VT52]:
;  275          1862  3              BEGIN
;  276          1863  3              CH$MOVE (2, VT52_LINE, .FREE_ADDR);
;  277          1864  3              .CUR_SIZE = ..CUR_SIZE + 2;
;  278          1865  2              END;
;  279          1866  2
;  280          1867  2          [VT100]:
;  281          1868  3              BEGIN
;  282          1869  3              CH$MOVE (3, VT100_LINE, .FREE_ADDR);
;  283          1870  3              .CUR_SIZE = ..CUR_SIZE + 3;
;  284          1871  2              END;
;  285          1872  2
;  286          1873  2          [HARDCOPY, UNKNOWN, VTFOREIGN]:
;  287          1874  2              ;
;  288          1875  2
;  289          1876  2          [INRANGE, OUTRANGE]:
;  290          1877  2              RETURN 0;                     ! should never get here
;  291          1878  2
;  292          1879  2      TES;
;  293          1880  2
;  294          1881  2      RETURN (SS$_NORMAL);
;  295          1882  2
;  296          1883  1      END;                                  ! End of routine COB$$ERASE_LINE_R2


                                                   0003E           .BLKB    2
                                      00  00  1E  00040 P.AAD:     .BYTE    30, 0, 0                                         ;
                                                   00043           .BLKB    1
                                          4B  1B  00044 P.AAE:     .BYTE    27, 75                                           ;
                                                   00046           .BLKB    2
                                      4B  5B  1B  00048 P.AAF:     .BYTE    27, 91, 75                                       ;

                                                        VT05_LINE=           P.AAD
                                                        VT52_LINE=           P.AAE
                                                        VT100_LINE=          P.AAF


                         51               62  C0 00000 COB$$ERASE_LINE_R2::
                                                                   ADDL2    (CUR_SIZE), FREE_ADDR                            ; 1851
                  05     00               50  CF 00003             CASEL    TERM_TYPE, #0, #5                                ; 1853
      001F      0016   000E             0028     00007 1$:         .WORD    6$-1$,-                                          ;
                       0028             0028     0000F                      2$-1$,-                                          ;
                                                                            3$-1$,-                                          ;
                                                                            4$-1$,-                                          ;
                                                                            6$-1$,-                                          ;
                                                                            6$-1$                                            ;
                                          1E  11 00013             BRB      7$                                               ; 1877
        61        18     00       DD  AF  F0 00015 2$:             INSV     VT05_LINE, #0, #24, (FREE_ADDR)                  ; 1857
                                          0F  11 0001B             BRB      5$                                               ; 1858
                         61       D9  AF  B0 0001D 3$:             MOVW     VT52_LINE, (FREE_ADDR)                           ; 1863
                         62           02  C0 00021                 ADDL2    #2, (CUR_SIZE)                                   ; 1864
                                      09  11 00024                 BRB      6$                                               ; 1853
        61          18          00      D4   AF  F0 00026 4$:      INSV    VT100_LINE, #0, #24, (FREE_ADDR)          ; 1869
                                62           03  C0 0002C 5$:      ADDL2   #3, (CUR_SIZE)                            ; 1870
                                50           01  D0 0002F 6$:      MOVL    #1, R0                                    ; 1881
                                                 05 00032          RSB                                               ;
                                             50  D4 00033 7$:      CLRL    R0                                        ; 1883
                                                 05 00035          RSB                                               ;

; Routine Size:  54 bytes,    Routine Base:  _COB$CODE + 004B

;  297          1884  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$ERASE_PAGE_R2 - Create erase page sequence'
1 GLOBAL ROUTINE COB$$ERASE_PAGE_R2 (
1                       TERM_TYPE,
1                       BUFFER,
1                       CUR_SIZE
1                   ) : COB$$ESC_R2_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence for erasing the
1 !       page from the current cursor position to the end of the
1 !       page.  The sequence is appended into the output buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$ERASE_PAGE_R2 (TERM_TYPE.rl.v,
1 !                                   BUFFER.mt.r, CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    LOCAL
2       FREE_ADDR;                      ! addr of next free byte in buffer
2
2    BIND
2       VT05_ERASE = UPLIT (BYTE (VT05_EOS, NULL, NULL)),
2       VT52_ERASE = UPLIT (BYTE (ESC, VT52_EOS)),
2       VT100_ERASE = UPLIT (BYTE (ESC, LB, VT100_EOS));
2
2    FREE_ADDR = .BUFFER + ..CUR_SIZE;
2
2    CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
2    SET
2       [VT05]:
3           BEGIN
3           CH$MOVE (3, VT05_ERASE, .FREE_ADDR);
```

```
;   356           1942  3                   .CUR_SIZE = ..CUR_SIZE + 3;
;   357           1943  2                   END;
;   358           1944  2
;   359           1945  2               [VT52]:
;   360           1946  3                   BEGIN
;   361           1947  3                   CH$MOVE (2, VT52_ERASE, .FREE_ADDR);
;   362           1948  3                   .CUR_SIZE = ..CUR_SIZE + 2;
;   363           1949  2                   END;
;   364           1950  2
;   365           1951  2               [VT100]:
;   366           1952  3                   BEGIN
;   367           1953  3                   CH$MOVE (3, VT100_ERASE, .FREE_ADDR);
;   368           1954  3                   .CUR_SIZE = ..CUR_SIZE + 3;
;   369           1955  2                   END;
;   370           1956  2
;   371           1957  2               [HARDCOPY, UNKNOWN, VTFOREIGN]:
;   372           1958  2                   ;
;   373           1959  2
;   374           1960  2               [INRANGE, OUTRANGE]:
;   375           1961  2                   RETURN 0;                       ! should never get here
;   376           1962  2
;   377           1963  2           TES;
;   378           1964  2
;   379           1965  2           RETURN (SS$_NORMAL);
;   380           1966  2
;   381           1967  1           END;                                    ! End of routine COB$$ERASE_PAGE_R2


                                                      00081          .BLKB   3
                                          00  00  1F  00084 P.AAG:   .BYTE   31, 0, 0                                      ;
                                                      00087          .BLKB   1
                                              4A  1B  00088 P.AAH:   .BYTE   27, 74                                        ;
                                                      0008A          .BLKB   2
                                          4A  5B  1B  0008C P.AAI:   .BYTE   27, 91, 74                                    ;

                                                            VT05_ERASE=        P.AAG
                                                            VT52_ERASE=        P.AAH
                                                            VT100_ERASE=       P.AAI


                              51              62  C0  00000 COB$$ERASE_PAGE_R2::
                                                                     ADDL2   (CUR_SIZE), FREE_ADDR                         ; 1935
                      05      00              50  CF  00003          CASEL   TERM_TYPE, #0, #5                             ; 1937
       001F         0016    000E            0028      00007 1$:      .WORD   6$-1$,-                                       ;
                            0028            0028      0000F                  2$-1$,-                                       ;
                                                                             3$-1$,-                                       ;
                                                                             4$-1$,-                                       ;
                                                                             6$-1$,-                                       ;
                                                                             6$-1$                                         ;
                                              1E  11  00013          BRB     7$                                            ; 1961
         61           18      00      DD      AF  F0  00015 2$:      INSV    VT05_ERASE, #0, #24, (FREE_ADDR)              ; 1941
                                              0F  11  0001B          BRB     5$                                            ; 1942
                              61      D9      AF  B0  0001D 3$:      MOVW    VT52_ERASE, (FREE_ADDR)                       ; 1947
                              62              02  C0  00021          ADDL2   #2, (CUR_SIZE)                                ; 1948
                                              09  11  00024          BRB     6$                                            ; 1937
         61           18      00      D4      AF  F0  00026 4$:      INSV    VT100_ERASE, #0, #24, (FREE_ADDR)             ; 1953
```

```
                 62          03  C0 0002C 5$:    ADDL2   #3, (CUR_SIZE)                     ; 1954
                 50          01  D0 0002F 6$:    MOVL    #1, R0                             ; 1965
                                 05 00032        RSB                                        ;
                             50  D4 00033 7$:    CLRL    R0                                 ; 1967
                                 05 00035        RSB                                        ;

; Routine Size:  54 bytes,    Routine Base:  _COB$CODE + 008F


;  382          1968  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$ERASE_WHOLE_LINE_R2 - Create erase whole line sequence'
1 GLOBAL ROUTINE COB$$ERASE_WHOLE_LINE_R2 (
1                       TERM_TYPE,
1                       BUFFER,
1                       CUR_SIZE
1                   ) : COB$$ESC_R2_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence to erase the entire
1 !       line containing the current cursor position. The string is
1 !       appended into the output buffer.
1 !
1 !       Notice that only VT100s have the ability to erase an entire
1 !       line regardless of whether the cursor is at the beginning
1 !       of that line.  Most terminals can only erase from the cursor
1 !       to the end of line.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$ERASE_WHOLE_LINE_R2 (TERM_TYPE.rl.v,
1 !                                                    BUFFER.mt.r,
1 !                                                    CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS.
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2   BEGIN
2
2   LOCAL
2       FREE_ADDR;                      ! addr of next free byte in buffer
2
2   BIND
2       VT05_LINE = UPLIT (BYTE (VT05_EOL, NULL, NULL)),
2       VT52_LINE = UPLIT (BYTE (ESC, VT52_EOL)),
2       VT100_WHOLE_LINE = UPLIT (BYTE (ESC, LB, TWO, VT100_EOL));
2
2     EE_ADDR = .BUFFER + ..CUR_SIZE;
```

```
;  441          2026  2
;  442          2027  2      CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
;  443          2028  2      SET
;  444          2029  2          [VT05]:
;  445          2030  3              BEGIN
;  446          2031  3              CH$MOVE (3, VT05_LINE, .FREE_ADDR);
;  447          2032  3              .CUR_SIZE = ..CUR_SIZE + 3;
;  448          2033  2              END;
;  449          2034  2
;  450          2035  2          [VT52]:
;  451          2036  3              BEGIN
;  452          2037  3              CH$MOVE (2, VT52_LINE, .FREE_ADDR);
;  453          2038  3              .CUR_SIZE = ..CUR_SIZE + 2;
;  454          2039  2              END;
;  455          2040  2
;  456          2041  2          [VT100]:
;  457          2042  3              BEGIN
;  458          2043  3              CH$MOVE (4, VT100_WHOLE_LINE, .FREE_ADDR);
;  459          2044  3              .CUR_SIZE = ..CUR_SIZE + 4;
;  460          2045  2              END;
;  461          2046  2
;  462          2047  2          [HARDCOPY, UNKNOWN, VTFOREIGN]:
;  463          2048  2              ;
;  464          2049  2
;  465          2050  2          [INRANGE, OUTRANGE]:
;  466          2051  2              RETURN 0;                     ! should never get here
;  467          2052  2
;  468          2053  2      TES;
;  469          2054  2
;  470          2055  2      RETURN (SS$_NORMAL);
;  471          2056  2
;  472          2057  1      END;                                  ! End of routine COB$$ERASE_WHOLE_LINE_R2


                                      000C5          .BLKB   3
                        00   00   1E  000C8 P.AAJ:   .BYTE   30, 0, 0                              ;
                                      000CB          .BLKB   1
                             4B   1B  000CC P.AAK:   .BYTE   27, 75                                ;
                                      000CE          .BLKB   2
                   4B   32   5B   1B  000D0 P.AAL:   .BYTE   27, 91, 50, 75                        ;

                                            VT05_LINE=          P.AAJ
                                            VT52_LINE=          P.AAK
                                            VT100_WHOLE_LINE=   P.AAL


                     51           62  C0 00000 COB$$ERASE_WHOLE_LINE_R2::
                                               ADDL2   @CUR_SIZE), FREE_ADDR                   ; 2025
                05   00           50  CF 00003 CASEL   TERM_TYPE, #0, #5                        ; 2027
    0022      0019   000E       0029     00007 1$:     .WORD   5$-1$,-                          ;
                     0029       0029     0000F                 2$-1$,-                          ;
                                                               3$-1$,-                          ;
                                                               4$-1$,-                          ;
                                                               5$-1$,-                          ;
                                                               5$-1$                            ;
                                  1F  11 00013         BRB     6$                               ; 2051
```

```
      61        18         00      DC   AF  F0 00015 2$:     INSV    VT05_LINE, #0, #24, (FREE_ADDR)           ; 2031
                           62           03  C0 0001B         ADDL2   #3, (CUR_SIZE)                            ; 2032
                           10           11  0001E            BRB     5$                                        ; 2027
                           61      D5   AF  B0 00020 3$:     MOVW    VT52_LINE, (FREE_ADDR)                    ; 2037
                           62           02  C0 00024         ADDL2   #2, (CUR_SIZE)                            ; 2038
                           07           11  00027            BRB     5$                                        ; 2027
                           61      D0   AF  D0 00029 4$:     MOVL    VT100_WHOLE_LINE, (FREE_ADDR)             ; 2043
                           62           04  C0 0002D         ADDL2   #4, (CUR_SIZE)                            ; 2044
                           50           01  D0 00030 5$:     MOVL    #1, R0                                    ; 2055
                                            05 00033         RSB                                               ;
                                        50  D4 00034 6$:     CLRL    R0                                        ; 2057
                                            05 00036         RSB                                               ;

; Routine Size:  55 bytes,    Routine Base:  _COB$CODE + 00D4

;  473          2058  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$ERASE_WHOLE_PAGE_R2 - Create erase whole page sequence'
1 GLOBAL ROU INE COB$$ERASE_WHOCE_PAGE_R2 (
1                     TERM_TYPE,
1                     BUFFER,
1                     CUR_SIZE
1                 ) : COB$$ESC_R2_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence to erase the
1 !       whole page regardless of cursor position.  The string is appended
1 !       into the output buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$ERASE_WHOLE_PAGE_R2 (TERM_TYPE.rl.v,
1 !                                                    BUFFER.mt.r,
1 !                                                    CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SICE EFFECTS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    LOCAL
2       FREE_ADDR;                          ! addr of next free byte in buffer
2
2    LITERAL
2       LINE1 = 32,                         ! 1 + 31 bias
2       COL1  = 32;                         ! 1 + 31 bias
2
2    BIND
2       VT05_ERASE = UPLIT (BYTE (VT05_EOS, NULL, NULL)),
2       VT52_^RASE = UPLIT (BYTE (ESC, VT52_SC, LINE1, COL1,
2                                 ESC, VT52_EOS)),
2       VT100_ιRASE_WHOLE = UPLIT (BYTE (ESC, LB, TWO, VT100_EOS));
2
2    FREE_ADDR = .BUFFER + ..CUR_SIZE;
```

```
;  532          2116  2
;  533          2117  2         CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
;  534          2118  2         SET
;  535          2119  2             [VT100]:
;  536          2120  3                 BEGIN
;  537          2121  3                 CH$MOVE (4, VT100_ERASE_WHOLE, .FREE_ADDR);
;  538          2122  3                 .CUR_SIZE = ..CUR_SIZE + 4;
;  539          2123  2                 END;
;  540          2124  2
;  541          2125  2             [VT52]:
;  542          2126  3                 BEGIN
;  543          2127  3                 !+
;  544          2128  3                 ! There is no sequence to erase the screen and leave the
;  545          2129  3                 ! cursor where it was, so on a VT52 we have to settle for
;  546          2130  3                 ! setting the cursor to 1,1 and erasing to the end of screen.
;  547          2131  3                 !-
;  548          2132  3                 CH$MOVE (6, VT52_ERASE, .FREE_ADDR);
;  549          2133  3                 .CUR_SIZE = ..CUR_SIZE + 6;
;  550          2134  2                 END;
;  551          2135  2
;  552          2136  2             [VT05]:
;  553          2137  3                 BEGIN
;  554          2138  3                 CH$MOVE (3, VT05_ERASE, .FREE_ADDR);
;  555          2139  3                 .CUR_SIZE = ..CUR_SIZE + 3;
;  556          2140  2                 END;
;  557          2141  2
;  558          2142  2             [HARDCOPY, UNKNOWN, VTFOREIGN]:
;  559          2143  2                 ;
;  560          2144  2
;  561          2145  2             [INRANGE, OUTRANGE]:
;  562          2146  2                 RETURN 0;                       ! should never get here
;  563          2147  2
;  564          2148  2         TES;
;  565          2149  2
;  566          2150  2         RETURN (SS$_NORMAL);
;  567          2151  2
;  568          2152  1         END;                                    ! End of routine COB$$ERASE_WHOLE_PAGE_R2


                                                 0010B          .BLKB    1
                                 00   00   1F    0010C P.AAM:   .BYTE    31, 0, 0                                   ;
                                                 0010F          .BLKB    1
                  4A   1B   20   20   59   1B    00110 P.AAN:   .BYTE    27, 89, 32, 32, 27, 74                     ;
                                                 00116          .BLKB    2
                            4A   32   5B   1B    00118 P.AAO:   .BYTE    27, 91, 50, 74                             ;

                                                       VT05_ERASE=          P.AAM
                                                       VT52_ERASE=          P.AAN
                                                       VT100_ERASE_WHOLE=   P.AAO


                               00F8    8F   BB 00000 COB$$ERASE_WHOLE_PAGE_R2::
                                                                PUSHR    #^M<R3,R4,R5,R6,R7>                        ; 2060
                      56               52   D0 00004            MOVL     R2, R6                                     ;
        57            51               66   C1 00007            ADDL3    (CUR_SIZE), BUFFER, FREE_ADDR              ; 2115
        05            00               50   CF 0000B            CASEL    TERM_TYPE, #0, #5                          ; 2117
        000E        0017        0021        002A    0000F 1$:   .WORD   5$-1$,-                                      ;
                                002A        002A    00017                4$-1$,-                                      ;
                                                                         3$-1$,-                                      ;
                                                                         2$-1$,-                                      ;
                                                                         5$-1$,-                                      ;
                                                                         5$-1$                                        ;
                                              21 11 0001B       BRB      6$                                           ; 2146
                                67     DC     AF D0 0001D 2$:   MOVL     VT100_ERASE_WHOLE, (FREE_ADDR)               ; 2121
                                66            04 C0 00021       ADDL2    #4, (CUR_SIZE)                               ; 2122
                                              13 11 00024       BRB      5$                                           ; 2117
                    67     CA   AF            06 28 00026 3$:   MOVC3    #6, VT52_ERASE, (FREE_ADDR)                  ; 2132
                                66            06 C0 0002B       ADDL2    #6, (CUR_SIZE)                               ; 2133
                                              09 11 0002E       BRB      5$                                           ; 2117
        67          18          00     BD     AF F0 00030 4$:   INSV     VT05_ERASE, #0, #24, (FREE_ADDR)             ; 2138
                                66            03 C0 00036       ADDL2    #3, (CUR_SIZE)                               ; 2139
                                50            01 D0 00039 5$:   MOVL     #1, R0                                       ; 2150
                                              02 11 0003C       BRB      7$                                           ;
                                              50 D4 0003E 6$:   CLRL     R0                                           ; 2152
                                     00F8     8F BA 00040 7$:   POPR     #^M<R3,R4,R5,R6,R7>                          ;
                                                 05 00044       RSB                                                   ;

; Routine Size:  69 bytes,    Routine Base:  _COB$CODE + 011C

;  569          2153  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$SET_ATTRIBUTES - Create set attributes sequence'
1 GLOBAL ROUTINE COB$$SET_ATTRIBUTES (
1                     TERM_TYPE,
1                     IN_TEXT,
1                     IN_LEN,
1                     FLAGS,
1                     OUT_BUF,
1                     OUT_LEN
1               ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence turning on
1 !       attributes such as bolding and blinking.  The attribute
1 !       sequence is placed in the output buffer, the input text
1 !       is copied over, and then the sequence to turn off graphics
1 !       is appended.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$SET_ATTRIBUTES (TERM_TYPE.rl.v, IN_TEXT.rt.r,
1 !                                              IN_LEN.rl.v, FLAGS.rl.v,
1 !                                              OUT_BUF.mt.r, OUT_LEN.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       IN_TEXT.rt.dx           descriptor of text which will have attr on
1 !       IN_LEN.rl.v             length of caller's text
1 !       FLAGS.rl.v              flags specifying which attributes to turn on
1 !       OUT_BUF.mt.r            addr of output buffer
1 !       OUT_LEN.ml.r            # bytes in output buffer, includes attributes,
1 !                                caller's text, & turn off graphic rendition
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    LOCAL
2       FREE_ADDR;
2
2    MACRO
2       VT100_OFF = %STRING (%CHAR (ESC), %CHAR (LB), '0m')%;
```

```
;  2
;  2      FREE_ADDR = .OUT_BUF + ..OUT_LEN;           ! init to first free byte
;  2
;  2      CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
;  2      SET
;  2          [HARDCOPY, UNKNOWN, VT05, VT52, VTFOREIGN]:
;  3              BEGIN
;  3              !+
;  3              ! Renditions not supported on these devices.  Just
;  3              ! copy the text into the output buffer and return.
;  3              !-
;  3              CH$MOVE (.IN_LEN, .IN_TEXT, .FREE_ADDR);
;  3              .OUT_LEN = ..OUT_LEN + .IN_LEN;
;  3              RETURN (SS$_NORMAL);
;  2              END;
;  2
;  2          [INRANGE, OUTRANGE]:
;  2              RETURN 0;                   ! error
;  2
;  2          [VT100]:
;  3              BEGIN
;  3              IF .FLAGS <0,4> EQL 0
;  3              THEN
;  4                  BEGIN                   ! no attr requested
;  4                  CH$MOVE (.IN_LEN, .IN_TEXT, .FREE_ADDR);
;  4                  .OUT_LEN = ..OUT_LEN + .IN_LEN;
;  4                  RETURN (SS$_NORMAL);
;  3                  END;
;  3              !+
;  3              ! For each attribute bit set in flags, copy
;  3              ! the appropriate ASCII graphic rendition byte
;  3              ! followed by a ';' into the output buffer.
;  3              ! Note use of autoincrementing.
;  3              !-
;  3              CH$WCHAR_A (ESC, FREE_ADDR);
;  3              CH$WCHAR_A (LB, FREE_ADDR);
;  3              INCR I FROM 0 TO 3
;  3              DO
;  4                  BEGIN                   ! build attribute string
;  4                  BIND
;  4                      ATTRTABL = UPLIT (BYTE ('1754')) : VECTOR [4, BYTE];
;  4
;  4                  IF .FLAGS <.I, 1>
;  4                  THEN
;  5                      BEGIN
;  5                      CH$WCHAR_A (.ATTRTABL [.I], FREE_ADDR);
;  5                      CH$WCHAR_A (%C';', FREE_ADDR);
;  5                      .OUT_LEN = ..OUT_LEN + 2; ! keep updating length
;  4                      END;
;  3                  END;
;  3
;  3              !+
;  3              ! When we fall out of above loop we have deposited
;  3              ! an extra ';' at the end of the buffer.  Back up
;  3              ! FREE_ADDR and write VT100_SGR on top of it.
;  3              !-
;  3              FREE_ADDR = .FREE_ADDR - 1;
;  685          2268  3              CH$WCHAR_A (VT100_SGR, FREE_ADDR);
;  686          2269  2              END;
;  687          2270  2          TES;
;  688          2271  2
;  689          2272  2      !+
;  690          2273  2      ! If we get here, the appropriate graphic rendition string has
;  691          2274  2      ! been moved to the output buffer.  Now copy the user's text over.
;  692          2275  2      !-
;  693          2276  2      FREE_ADDR = CH$MOVE (.IN_LEN, .IN_TEXT, .FREE_ADDR);
;  694          2277  2
;  695          2278  2      !+
;  696          2279  2      ! Append in sequence to turn off graphic rendition.
;  697          2280  2      !-
;  698          2281  2      CH$MOVE (%CHARCOUNT (VT100_OFF), UPLIT (BYTE (VT100_OFF)), .FREE_ADDR);
;  699          2282  2
;  700          2283  2      !+
;  701          2284  2      ! Set the output length and exit.
;  702          2285  2      !-
;  703          2286  2      .OUT_LEN = ..OUT_LEN + .IN_LEN + 6;  ! add length of caller's text &
;  704          2287  2                                           !  turn on/off graphic rendition
;  705          2288  2      RETURN (SS$_NORMAL);
;  706          2289  2
;  707          2290  1      END;                                 ! End of routine COB$$SET_ATTRIBUTES



                                                      00161              .BLKB   3
                                  34  35  37  31      00164 P.AAP:       .ASCII  \1754\                                          ;
                                  6D  30  5B  1B      00168 P.AAQ:       .ASCII  <27>\[0m\                                       ;

                                                            ATTRTABL=            P.AAP


                                            00FC      00000              .ENTRY  COB$$SET_ATTRIBUTES, Save R2,R3,R4,R5,R6,R7  ; 2155
                          56          18    AC  D0    00002              MOVL    OUT_LEN, R6                                  ; 2212
           57        14   AC                66  C1    00006              ADDL3   (R6), OUT_BUF, FREE_ADDR                     ;
           05             00          04    AC  CF    0000B              CASEL   TERM_TYPE, #0, #5                            ; 2214
000E     0014           0014          0014            00010 1$:          .WORD   3$-1$,-                                      ;
                        0014          0014            00018                      3$-1$,-                                      ;
                                                                                 3$-1$,-                                      ;
                                                                                 2$-1$,-                                      ;
                                                                                 3$-1$,-                                      ;
                                                                                 3$-1$                                        ;
                                            4D  11    0001C              BRB     8$                                           ; 2228
                          0F          10    AC  93    0001E 2$:          BITB    FLAGS, #15                                   ; 2232
                                            0C  12    00022              BNEQ    4$                                           ;
           67        08   BC          0C    AC  28    00024 3$:          MOVC3   IN_LEN, @IN_TEXT, (FREE_ADDR)                ; 2235
                          66          0C    AC  C0    0002A              ADDL2   IN_LEN, (R6)                                 ; 2236
                                            37  11    0002E              BRB     7$                                           ; 2237
                          87        5B1B    8F  B0    00030 4$:          MOVW    #23323, (FREE_ADDR)+                         ; 2245
                                            50  D4    00035              CLRL    I                                            ; 2247
           0B        10   AC                50  E1    00037 5$:          BBC     I, FLAGS, 6$                                 ; 2253
                          87          B8  AF40  90    0003C              MOVB    ATTRTABL[I], (FREE_ADDR)+                    ; 2256
                          87                3B  90    00041              MOVB    #59, (FREE_ADDR)+                            ; 2257
                          66                02  C0    00044              ADDL2   #2, (R6)                                     ; 2258
           EC             50                03  F3    00047 6$:          AOBLEQ  #3, I, 5$                                    ; 2247
                          77          6D    8F  90    0004B              MOVB    #109, -(FREE_ADDR)                           ; 2268
```

```
                                        57  D6 0004F          INCL    FREE_ADDR                                 :
                  67       08   BC     0C   AC  28 00051          MOVC3   IN_LEN, @IN_TEXT, (FREE_ADDR)             : 2276
                                57          53  D0 00057          MOVL    R3, FREE_ADDR                             :
                                67     9F   AF  D0 0005A          MOVL    P.AAQ, (FREE_ADDR)                        : 2281
                  50            66     0C   AC  C1 0005E          ADDL3   IN_LEN, (R6), R0                          : 2286
                                66     06   A0  9E 00063          MOVAB   6(R0), (R6)                               :
                                50          01  D0 00067 7$:      MOVL    #1, R0                                    : 2288
                                                04 0006A          RET                                               :
                                            50  D4 0006B 8$:      CLRL    R0                                        : 2290
                                                04 0006D          RET                                               :

; Routine Size:  110 bytes,    Routine Base:  _COB$CODE + 016C


;  708          2291  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$SET_ATTRIBUTES_ONLY - Create only set attributes sequence'
1 GLOBAL ROUTINE COB$$SET_ATTRIBUTES_ONLY (
1                       TERM_TYPE,
1                       FLAGS,
1                       PREFIX_BUF,
1                       P_PREFIX_LEN,
1                       SUFFIX_BUF,
1                       P_SUFFIX_LEN
1                   ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequences turning on and off
1 !       attributes such as bolding and blinking.  These attribute
1 !       sequences are placed in two buffers supplied by the caller.
1 !       No input text is specified.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$SET_ATTRIBUTES (TERM_TYPE.rl.v,
1 !                                               FLAGS.rl.v,
1 !                                               PREFIX_BUF.mt.r,
1 !                                               P_PREFIX_LEN.ml.r,
1 !                                               SUFFIX_BUF.mt.r,
1 !                                               P_SUFFIX_LEN.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       FLAGS.rl.v              flags specifying which attributes to turn on
1 !       PREFIX_BUF.mt.r         addr of output buffer to receive prefix string
1 !       P_PREFIX_LEN.ml.r       # bytes in already in prefix buffer
1 !                               gets updated to include size of prefix
1 !       SUFFIX_BUF.mt.r         addr of output buffer to receive suffix string
1 !       P_SUFFIX_LEN.ml.r       # bytes in already in suffix buffer
1 !                               gets updated to include size of suffix
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
```

```
2 BEGIN
2
2 BIND
2
2           PREFIX_LEN      = .P_PREFIX_LEN,         ! holds length of prefix buffer
2           SUFFIX_LEN      = .P_SUFFIX_LEN;         ! holds length of suffix buffer
2
2 LOCAL
2
2           BUFFER_PTR;
2
2 MACRO
2
2           VT100_OFF = %STRING (%CHAR (ESC), %CHAR (LB), '0m')%;
2
2 BUFFER_PTR = .PREFIX_BUF + .PREFIX_LEN; ! init to first free byte of prefix
2
2 CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
2           SET
2           [HARDCOPY, UNKNOWN, VT05, VT52, VTFOREIGN]:
3               BEGIN
3               !+
3               ! Renditions not supported on these devices.  Just return.
3               !-
3               RETURN      SS$_NORMAL
2               END;
2
2           [INRANGE, OUTRANGE]:
2               RETURN 0;                    ! error
2
2           [VT100]:
3               BEGIN
3               IF .FLAGS <0,4> EQL 0
3               THEN
3                   RETURN (SS$_NORMAL);     ! no attributes requested
3
3               !+
3               ! For each attribute bit set in flags, copy
3               ! the appropriate ASCII graphic rendition byte
3               ! followed by a ';' into the output buffer.
3               ! Note use of autoincrementing.
3               !-
3
3               CH$WCHAR_A (ESC, BUFFER_PTR);
3               CH$WCHAR_A (LB,  BUFFER_PTR);
3               PREFIX_LEN = .PREFIX_LEN + 2; ! Start with 2 chars: <ESC> "["
3               INCR I FROM 0 TO 3
3               DO
4                   BEGIN                    ! build prefix attribute string
4                   BIND
4                       ATTRTABL = UPLIT (BYTE ('1754')) : VECTOR [4, BYTE];
4
4                   IF .FLAGS <.I, 1>
4                   THEN
5                       BEGIN
5                       CH$WCHAR_A (.ATTRTABL[.I], BUFFER_PTR);
5                       CH$WCHAR_A (%C';', BUFFER_PTR);
```

```
;  820          2401  5                     PREFIX_LEN = .PREFIX_LEN + 2; ! keep updating length
;  821          2402  4                     END;
;  822          2403  3                 END;                         ! build prefix attribute string
;  823          2404  3
;  824          2405  3             !+
;  825          2406  3             ! When we fall out of above loop we have deposited
;  826          2407  3             ! an extra ';' at the end of the buffer.  Back up
;  827          2408  3             ! FREE_ADDR and write VT100_SGR on top of it.
;  828          2409  3             !-
;  829          2410  3             BUFFER_PTR = .BUFFER_PTR - 1;
;  830          2411  3             CH$WCHAR_A (VT100_SGR, BUFFER_PTR);
;  831          2412  3
;  832          2413  2             END;
;  833          2414  2         TES;
;  834          2415  2
;  835          2416  2 !+
;  836          2417  2 ! Append in sequence to turn off graphic rendition.
;  837          2418  2 !-
;  838          2419  2
;  839          2420  2 BUFFER_PTR = .SUFFIX_BUF + .SUFFIX_LEN; ! init to first free byte in
;  840          2421  2                                       ! suffix buffer.
;  841          2422  2
;  842          2423  2 CH$MOVE (%CHARCOUNT (VT100_OFF), UPLIT (BYTE (VT100_OFF)), .BUFFER_PTR);
;  843          2424  2
;  844          2425  2 !+
;  845          2426  2 ! Set the output length and exit.
;  846          2427  2 !-
;  847          2428  2
;  848          2429  2 SUFFIX_LEN = .SUFFIX_LEN + %CHARCOUNT(VT100_OFF);
;  849          2430  2
;  850          2431  2 RETURN  SS$_NORMAL
;  851          2432  2
;  852          2433  1 END;                             ! End of routine COB$$SET_ATTRIBUTES_ONLY


                                            001DA         .BLKB   2
                               34  35  37  31 001DC P.AAR:  .ASCII  \1754\                                  ;
                               6D  30  5B  1B 001E0 P.AAS:  .ASCII  <27>\[0m\                               ;

                                                  ATTRTABL=         P.AAR


                                          0004 00000        .ENTRY  COB$$SET_ATTRIBUTES_ONLY, Save R2      ; 2293
                            52        10  AC  D0 00002        MOVL    P PREFIX_LEN, R2                      ; 2348
                   51      0C  AC          62  C1 00006        ADDL3   (R2), PREFIX_BUF, BUFFER_PTR          ; 2359
                   05          00      04  AC  CF 0000B        CASEL   TERM_TYPE, #0, #5                     ; 2361
       000E       0046       0046          0046     00010 1$:  .WORD   5$-1$,-                               ;
                             0046          0046     00018              5$-1$,-                               ;
                                                                       5$-1$,-                               ;
                                                                       2$-1$,-                               ;
                                                                       5$-1$,-                               ;
                                                                       5$-1$                                 ;
                                               3C  11 0001C        BRB     6$                                    ; 2372
                            0F        08  AC  93 0001E 2$:     BITB    FLAGS, #15                            ; 2376
                                          32  13 00022        BEQL    5$                                    ;
                            81      5B1B  8F  B0 00024        MOVW    #23323, (BUFFER_PTR)+                 ; 2387
```

```
                          62           02  C0 00029      ADDL2   #2, (R2)                                        ; 2389
                                       50  D4 0002C      CLRL    I                                               ; 2390
              0B     08   AC           50  E1 0002E 3$:  BBC     I, FLAGS, 4$                                    ; 2396
                          81     C1 AF40   90 00033      MOVB    ATTRTABL[I], (BUFFER_PTR)+                      ; 2399
                          81           3B  90 00038      MOVB    #59, (BUFFER_PTR)+                              ; 2400
                          62           02  C0 0003B      ADDL2   #2, (R2)                                        ; 2401
              EC          50           03  F3 0003E 4$:  AOBLEQ  #3, I, 3$                                       ; 2390
                          71       6D  8F  90 00042      MOVB    #109, -(BUFFER_PTR)                             ; 2411
                                       51  D6 00046      INCL    BUFFER_PTR                                      ;
              51     14   AC       18  BC  C1 00048      ADDL3   @P_SUFFIX_LEN, SUFFIX_BUF, BUFFER_PTR           ; 2420
                          61       AB  AF  D0 0004E      MOVL    P.AAS, (BUFFER_PTR)                             ; 2423
                     18   BC           04  C0 00052      ADDL2   #4, @P_SUFFIX_LEN                               ; 2429
                          50           01  D0 00056 5$:  MOVL    #1, R0                                          ; 2431
                                           04 00059      RET                                                     ;
                                       50  D4 0005A 6$:  CLRL    R0                                              ; 2433
                                           04 0005C      RET                                                     ;

; Routine Size:  93 bytes,    Routine Base:  _COB$CODE + 01E4


;  853          2434  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$SET_CURSOR_ABS_R4 - Create absolute set cursor sequence'
1 GLOBAL ROUTINE COB$$SET_CURSOR_ABS_R4 (
1                           TERM_TYPE,
1                           LINE_NO,
1                           COL_NO,
1                           BUFFER,
1                           CUR_SIZE
1                       ) : COB$$ESC_R4_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence for a set cursor
1 !       position and appends the string to a given output buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$SET_CURSOR_ABS_R4 (TERM_TYPE.rl.v, LINE_NO.rl.v,
1 !                                                   COL_NO.rl.v, BUFFER.mt.r,
1 !                                                   CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       LINE_NO.rl.v            line number
1 !       COL_NO.rl.v             column number
1 !       BUFFER.mt.r             addr of buffer
1 !                                this buffer should be at least
1 !                                20 bytes
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2    BEGIN
2
2    LOCAL
2       VT100CTL : VECTOR [1, 8] INITIAL (
2           DSC$K_CLASS_S ^24 + DSC$K_DTYPE_T ^16 + 10,
2           UPLIT ( BYTE (ESC, LB, '!OL;!UL', VT100_SC ))),
2                                       ! dsc for cvt to vt100 sequence
2                                       ! FAO control string
2       FREE_ADDR : REF VECTOR [,BYTE]; ! addr of 1st free byte
2
```

```
;  912          2492  2      FREE_ADDR = .BUFFER + ..CUR_SIZE;    ! addr of next free byte
;  913          2493  2
;  914          2494  2      CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
;  915          2495  2          SET
;  916          2496  2
;  917          2497  2          [HARDCOPY, UNKNOWN, VTFOREIGN]:
;  918          2498  2              ;                           ! do nothing
;  919          2499  2
;  920          2500  2          [VT05]:
;  921          2501  3              BEGIN
;  922          2502  3              .CUR_SIZE = ..CUR_SIZE + 3; ! update current size of buffer
;  923          2503  3              FREE_ADDR [0] = VT05_SC;    ! put set cursor sequence into buffer
;  924          2504  3              FREE_ADDR [1] = CB + .LINE_NO;
;  925          2505  3              FREE_ADDR [2] = CB + .COL_NO;
;  926          2506  2              END;
;  927          2507  2
;  928          2508  2          [VT52]:
;  929          2509  3              BEGIN
;  930          2510  3              .CUR_SIZE = ..CUR_SIZE + 4; ! update current size of buffer
;  931          2511  3              FREE_ADDR [0] = ESC;        ! put set cursor sequence into buffer
;  932          2512  3              FREE_ADDR [1] = VT52_SC;
;  933          2513  3              FREE_ADDR [2] = CB + .LINE_NO;
;  934          2514  3              FREE_ADDR [3] = CB + .COL_NO;
;  935          2515  2              END;
;  936          2516  2
;  937          2517  2          [VT100]:
;  938          2518  3              BEGIN
;  939          2519  3              LOCAL
;  940          2520  3                  STATUS,
;  941          2521  3                  CVT_ARGS : VECTOR [2],
;  942          2522  3                  FAO_BUFFER : BLOCK [8, BYTE],
;  943          2523  3                  FAO_LEN : WORD;
;  944          2524  3
;  945          2525  3              CVT_ARGS [0] = .LINE_NO;
;  946          2526  3              CVT_ARGS [1] = .COL_NO;
;  947          2527  3              FAO_BUFFER [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  948          2528  3              FAO_BUFFER [DSC$B_CLASS] = DSC$K_CLASS_S;
;  949          2529  3              FAO_BUFFER [DSC$W_LENGTH] = 20;                ! arbitrary - sb large enough
;  950          2530  3              FAO_BUFFER [DSC$A_POINTER] = .FREE_ADDR;
;  951          2531  3
;  952          2532  3              !+
;  953          2533  3              ! Convert to ASCII characters and move to buffer.
;  954          2534  3              !-
;  955        P 2535  3              STATUS = $FAOL (CTRSTR = VT100CTL, OUTLEN = FAO_LEN,
;  956          2536  3                  OUTBUF = FAO_BUFFER, PRMLST = CVT_ARGS);
;  957          2537  3              IF NOT .STATUS THEN RETURN (.STATUS);
;  958          2538  3              .CUR_SIZE = ..CUR_SIZE + .FAO_LEN;  ! add length of appended string
;  959          2539  3
;  960          2540  2              END;
;  961          2541  2
;  962          2542  2          [INRANGE,OUTRANGE]:
;  963          2543  2              RETURN 0;                   ! should never get here
;  964          2544  2
;  965          2545  2          TES;
;  966          2546  2
;  967          2547  2      RETURN 1;
;  968          2548  2
;  969          2549  1      END;                                ! End of routine COB$$SET_CURSOR_ABS_R4


                                                00241 P.AAT:    .BLKB    3
                                      5B   1B   00244           .BYTE    27, 91                                   :
                    4C   55   21   3B   4C   55   21   00246    .ASCII   \!UL;!UL\                                :
                                           66   0024D           .BYTE    102                                      :

                                                                .EXTRN   SYS$FAOL

                           5E        1C   C2 00000 COB$$SET_CURSOR_ABS_R4::
                                                                SUBL2    #28, SP                                  : 2436
                    14   AE 010E000A   8F   D0 00003            MOVL     #17694730, VT100CTL                      : 2481
                    18   AE       E8   AF   9E 0000B            MOVAB    P.AAT, VT100CTL+4                        :
                         53            64   C0 00010            ADDL2    (CUR_SIZE), FREE_ADDR                    : 2492
               05        00            50   CF 00013            CASEL    TERM_TYPE, #0, #5                        : 2494
   0034        0020    000E          0060      00017 1$:        .WORD    5$-1$,-                                  :
                       0060          0060      0001F                     2$-1$,-                                  :
                                                                         3$-1$,-                                  :
                                                                         4$-1$,-                                  :
                                                                         5$-1$,-                                  :
                                                                         5$-1$                                    :
                                       57   11 00023            BRB      6$                                       : 2543
                         64            03   C0 00025 2$:        ADDL2    #3, (CUR_SIZE)                           : 2502
                         63            0E   90 00028            MOVB     #14, (FREE_ADDR)                         : 2503
              01   A3    51            1F   81 0002B            ADDB3    #31, LINE_NO, 1(FREE_ADDR)               : 2504
              02   A3    52            1F   81 00030            ADDB3    #31, COL_NO, 2(FREE_ADDR)                : 2505
                                       40   11 00035            BRB      5$                                       : 2494
                         64            04   C0 00037 3$:        ADDL2    #4, (CUR_SIZE)                           : 2510
                         63     591B   8F   B0 0003A            MOVW     #22811, (FREE_ADDR)                      : 2511
              02   A3    51            1F   81 0003F            ADDB3    #31, LINE_NO, 2(FREE_ADDR)               : 2513
              03   A3    52            1F   81 00044            ADDB3    #31, COL_NO, 3(FREE_ADDR)                : 2514
                                       2C   11 00049            BRB      5$                                       : 2494
                    0C   AE            51   7D 0004B 4$:        MOVQ     LINE_NO, CVT_ARGS                        : 2525
                    04   AE 010E0014   8F   D0 0004F            MOVL     #17694740, FAO_BUFFER                    : 2529
                    08   AE            53   D0 00057            MOVL     FREE_ADDR, FAO_BUFFER+4                  : 2530
                                  0C   AE   9F 0005B            PUSHAB   CVT_ARGS                                 : 2536
                                  08   AE   9F 0005E            PUSHAB   FAO_BUFFER                               :
                                  08   AE   9F 00061            PUSHAB   FAO_LEN                                  :
                                  20   AE   9F 00064            PUSHAB   VT100CTL                                 :
                  00000000G   00           04   FB 00067            CALLS    #4, SYS$FAOL                             :
                         0D            50   E9 0006E            BLBC     STATUS, 7$                               : 2537
                         50            6E   3C 00071            MOVZWL   FAO_LEN, R0                              : 2538
                         64            50   C0 00074            ADDL2    R0, (CUR_SIZE)                           :
                         50            01   D0 00077 5$:        MOVL     #1, R0                                   : 2547
                                       02   11 0007A            BRB      7$                                       :
                                       50   D4 0007C 6$:        CLRL     R0                                       : 2549
                         5E            1C   C0 0007E 7$:        ADDL2    #28, SP                                  :
                                            05 00081            RSB                                               :

; Routine Size:  130 bytes,    Routine Base:  _COB$CODE + 024E


;  970          2550  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$SET_CURSOR_REL Create relative cursor position sequence'
1 GLOBAL ROUTINE COB$$SET_CURSOR_REL (
1                       TERM_TYPE,
1                       LINE_NO,
1                       COL_NO,
1                       LINE_PLUS,
1                       COL_PLUS,
1                       BUFFER,
1                       CUR_SIZE
1                   ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence to position
1 !       the cursor relative to the specified line and column, or
1 !       relative to the current position if none is specified.
1 !       The set cursor sequence is appended to the output string.
1 !
1 !       Notice that the ANSI sequences can become quite large.
1 !       For instance, it is possible that 50 up arrows (2 bytes each)
1 !       will be only a part of the resulting sequence.  It is
1 !       recommended that the output buffer be 512 bytes long.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$SET_CURSOR_REL (TERM_TYPE.rl.v, LINE_NO.rl.v,
1 !                                  COL_NO.rl.v, LINE_PLUS.rl.v,
1 !                                  COL_PLUS.rl.v, BUFFER.mt.r,
1 !                                  CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       LINE_NO.rl.v            line number
1 !       COL_NO.rl.v             column number
1 !       LINE_PLUS.rl.v          offset from line number
1 !       COL_PLUS.rl.v           offset from column number
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1 !+
1 !  The following table shows the cursor positioning used for every
```

```
1 !  combination of the LINE and COLUMN phrases on both ANSI devices
1 !  and VT100s.  The arrows on the VT52 can only be moved one position at
1 !  a  time.  This may be slower, but at least the results will be the
1 !  same as far as cursor positioning goes on both types of terminals.
1 !
1 !   "v" = down arrow
1 !
1 !   "^" = up arrow
1 !
1 !   LINE a ! LINE PLUS b ! COLUMN c ! COLUMN PLUS d !  Cursor Pos. Used
1 !   ------------------------------------------------------------------
1 !          !             !          !               !
1 !     N    !      N      !     N    !       N       ! Current   Rules
1 !     N    !      N      !     N    !       Y       ! d "->"
1 !     N    !      N      !     Y    !       N       ! <CR> ; c-1 "->"
1 !     N    !      N      !     Y    !       Y       ! <CR> ; (c-1)+d "->"
1 !     N    !      Y      !     N    !       N       ! b <LF>
1 !     N    !      Y      !     N    !       Y       ! b <LF> ; d "->"
1 !     N    !      Y      !     Y    !       N       ! b <LF> ; <CR> ; c-1 "->"
1 !     N    !      Y      !     Y    !       Y       ! b <LF> ; <CR> ; (c-1)+d "->"
1 !     Y    !      N      !     N    !       N       ! Home ; a-1 "v"
1 !     Y    !      N      !     N    !       Y       ! 24 "^" ; a-1 "v" ; d "->"
1 !     Y    !      N      !     Y    !       N       ! Direct a,c
1 !     Y    !      N      !     Y    !       Y       ! Direct a,c+d
1 !     Y    !      Y      !     N    !       N       ! Home ; a-1 "v" ; b "LF"
1 !     Y    !      Y      !     N    !       Y       ! 24 "^" ; a-1 "v" ; b <LF>
1 !          !             !          !               !    d "->"
1 !     Y    !      Y      !     Y    !       N       ! Direct a,c ; b <LF>
1 !     Y    !      Y      !     Y    !       Y       ! Direct a,c+d ; b <LF>
1 !          !             !          !               !
1 !    ------------------------------------------------------------------
1 !
1 !     note: <lf> for all LINE PLUS to get scrolling
1 !     note: 24 up arrows used instead of home - this maintains the current
1 !     column position
1 !-
```

```
  1
  2      BEGIN
  2
  2      !+
  2      ! The following macro will put the VT100 sequence for
  2      ! multiple arrow movement into the buffer and update
  2      ! the length and pointer.  Sequences are of the form
  2      ! ESC [ num arrow.
  2      !-
  2      MACRO
M 2          $APPEND_VT100_SEQ (NUM, CTR_ARROW) =
M 2          BEGIN
M 2          LOCAL
M 2              CVT_ARG,
M 2              FAO_BUF : BLOCK [8, BYTE],
M 2              FAO_LEN : WORD,
M 2              STATUS;
M 2
M 2          IF NUM NEQ 0
M 2          THEN
M 2              BEGIN
M 2              CVT_ARG = NUM;
M 2              FAO_BUF [DSC$B_DTYPE] = DSC$K_DTYPE_T;
M 2              FAO_BUF [DSC$B_CLASS] = DSC$K_CLASS_S;
M 2              FAO_BUF [DSC$W_LENGTH] = 15;              ! arbitrary - sb big enough
M 2              FAO_BUF [DSC$A_POINTER] = .FREE_ADDR;
M 2
M 2              STATUS = $FAOL (CTRSTR = CTR_ARROW, OUTLEN   FAO_LEN,
M 2                          OUTBUF = FAO_BUF, PRMLST = CVT_ARG);
M 2              IF NOT .STATUS THEN RETURN .STATUS;
M 2
M 2              .CUR_SIZE = ..CUR_SIZE + .FAO_LEN;
M 2              FREE_ADDR = .FREE_ADDR + .FAO_LEN;
M 2              END;
M 2          END
  2  %;                                    ! end macro $append_vt100_seq
  2
  2      !+
  2      ! This macro puts NUM arrows into the buffer.
  2      ! The next free byte and buffer size are updated.
  2      !-
  2      MACRO
M 2          $APPEND_N_ARROWS (NUM, DIRECTION) =
M 2          BEGIN
M 2          INCR COUNTER FROM 1 TO NUM DO
M 2              BEGIN
M 2              FREE_ADDR = CH$MOVE (2, UPLIT (BYTE (ESC, DIRECTION)), .FREE_ADDR);
M 2              .CUR_SIZE = ..CUR_SIZE + 2;
M 2              END;
M 2          END;
  2  %;                                    ! end of macro append_n_arrows
  2
  2      MACRO
M 2          $APPEND_VT100_HOME =
M 2          BEGIN
M 2          FREE_ADDR = CH$MOVE (3, UPLIT (BYTE( ESC, LB, f)),
M 2                                  .FREE_ADDR);
```

```
; 1123        M 2701  2               .CUR_SIZE = ..CUR_SIZE + 3;
; 1124        M 2702  2               END
; 1125          2703  2 %;
; 1126          2704  2
; 1127          2705  2         MACRO
; 1128        M 2706  2             $APPEND_VT52_HOME =
; 1129        M 2707  2             BEGIN
; 1130        M 2708  2             FREE_ADDR = CH$MOVE (2, UPLIT (BYTE (ESC, H)), .FREE_ADDR);
; 1131        M 2709  2             .CUR_SIZE = ..CUR_SIZE + 2;
; 1132        M 2710  2             END;
; 1133          2711  2 %;
; 1134          2712  2
; 1135          2713  2         LOCAL
; 1136          2714  2             FREE_ADDR : REF VECTOR [,BYTE],
; 1137          2715  2             UP_CTL : VECTOR [1, 8] INITIAL (
; 1138          2716  2                     DSC$K_CLASS_S ^ 24 + DSC$K_DTYPE_T ^ 16 + 6,
; 1139          2717  2                     UPLIT (BYTE (ESC, LB, '!UL', A))),
; 1140          2718  2             DOWN_CTL : VECTOR [1, 8] INITIAL (
; 1141          2719  2                     DSC$K_CLASS_S ^ 24 + DSC$K_DTYPE_T ^ 16 + 6,
; 1142          2720  2                     UPLIT (BYTE (ESC, LB, '!UL', B))),
; 1143          2721  2             RIGHT_CTL : VECTOR [1, 8] INITIAL (
; 1144          2722  2                     DSC$K_CLASS_S ^ 24 + DSC$K_DTYPE_T ^ 16 + 6,
; 1145          2723  2                     UPLIT (BYTE (ESC, LB, '!UL', C)));
; 1146          2724  2
; 1147          2725  2         BIND
; 1148          2726  2             UP = A,                             ! equate letters to directions
; 1149          2727  2             DOWN = B,
; 1150          2728  2             RIGHT = C;
; 1151          2729  2
; 1152          2730  2         LITERAL
; 1153          2731  2             K_MAX_RMS_SIZE = 255;
```

```
2      IF .TERM_TYPE NEQ VT100 AND
2          .TERM_TYPE NEQ VT52
2      THEN RETURN (SS$_NORMAL);              ! don't do anything for other
2                                             !  terminal types
2
2      FREE_ADDR = .BUFFER + ..CUR_SIZE;
2
2      IF .LINE_NO NEQ 0 AND
2          .COL_NO NEQ 0
2      THEN                                   ! direct cursor addressing
3          BEGIN
3          COB$$SET_CURSOR_ABS_R4 (.TERM_TYPE, .LINE_NO,
3                                  .COL_NO + .COL_PLUS, .BUFFER,
3                                  .CUR_SIZE);
3          FREE_ADDR = .BUFFER + ..CUR_SIZE; ! update addr next free byte
2          END;
2
2      IF .LINE_NO NEQ 0 AND
2          .COL_NO EQL 0
2      THEN
3          BEGIN
3          IF .COL_PLUS EQL 0
3          THEN                        ! insert home sequence
4              BEGIN
4              IF .TERM_TYPE EQL VT100
4              THEN
5                  $APPEND_VT100_HOME
4              ELSE
4                  $APPEND_VT52_HOME;
4              END
3          ELSE
4              BEGIN                   ! insert a bunch of up arrows
4              MACRO
4                  UP_ARROW = %STRING (%CHAR (ESC), %CHAR (A))%;
4              BIND
4                  UP_24 = UPLIT (BYTE (REP 24 OF (UP_ARROW)));
4
4              IF .TERM_TYPE EQL VT100
4              THEN
5                  $APPEND_VT100_SEQ (24, UP_CTL)
4              ELSE
5                  BEGIN
5                  FREE_ADDR = CH$MOVE (48, UP_24, .FREE_ADDR);
5                  .CUR_SIZE = ..CUR_SIZE + 48;
4                  END;
3              END;
3          !+
3          ! Insert line_no down arrows regardless of col_plus
3          !-
3          IF .TERM_TYPE EQL VT100
3          THEN
4              $APPEND_VT100_SEQ (.LINE_NO - 1, DOWN_CTL)
3          ELSE
3              $APPEND_N_ARROWS (.LINE_NO - 1, DOWN);
2          END;
2
2      IF .LINE_NO EQL 0 AND
```

```
; 1212          2789 2           .COL_NO NEQ 0
; 1213          2790 2       THEN                                ! insert a CR &
; 1214          2791 3           BEGIN                           ! col_no right arrows
; 1215          2792 3           FREE_ADDR [0] = CR;
; 1216          2793 3           FREE_ADDR = .FREE_ADDR + 1;
; 1217          2794 3           .CUR_SIZE = ..CUR_SIZE + 1;
; 1218          2795 2           END;
; 1219          2796 2
; 1220          2797 2       IF .LINE_PLUS NEQ 0
; 1221          2798 2       THEN                                ! add line_plus LFs to buffer
; 1222          2799 3           BEGIN
; 1223          2800 3           FREE_ADDR = CH$FILL (LF, .LINE_PLUS, .FREE_ADDR);
; 1224          2801 3           .CUR_SIZE = ..CUR_SIZE + .LINE_PLUS;
; 1225          2802 2           END;
; 1226          2803 2
; 1227          2804 2       IF (.COL_PLUS NEQ 0 OR .COL_NO NEQ 0) AND
; 1228          2805 3          (.LINE_NO EQL 0 OR .COL_NO EQL 0) ! didn't do direct cursor addr
; 1229          2806 2       THEN                                ! insert col_plus right arrows
; 1230          2807 3           BEGIN
; 1231          2808 3           LOCAL
; 1232          2809 3               COL;
; 1233          2810 3           COL = .COL_NO - 1;
; 1234          2811 3           IF .COL LSS 0
; 1235          2812 3           THEN
; 1236          2813 3               COL = 0;
; 1237          2814 3           IF .TERM_TYPE EQL VT100
; 1238          2815 3           THEN
; 1239          2816 4               $APPEND_VT100_SEQ (.COL + .COL_PLUS, RIGHT_CTL)
; 1240          2817 3           ELSE
; 1241          2818 3               $APPEND_N_ARROWS (.COL + .COL_PLUS, RIGHT);
; 1242          2819 2           END;
; 1243          2820 2
; 1244          2821 2       RETURN (SS$_NORMAL);                ! everything should be in the buffer
; 1245          2822 2
; 1246          2823 1       END;                                ! End of routine COB$$SET_CURSOR_REL


                                    5B  1B  002D0 P.AAU:  .BYTE   27, 91                              ;
                                4C  55  21  002D2         .ASCII  \!UL\                               ;
                                        41  002D5         .BYTE   65                                  ;
                                            002D6         .BLKB   2
                                    5B  1B  002D8 P.AAV:  .BYTE   27, 91                              ;
                                4C  55  21  002DA         .ASCII  \!UL\                               ;
                                        42  002DD         .BYTE   66                                  ;
                                            002DE         .BLKB   2
                                    5B  1B  002E0 P.AAW:  .BYTE   27, 91                              ;
                                4C  55  21  002E2         .ASCII  \!UL\                               ;
                                        43  002E5         .BYTE   67                                  ;
                                            002E6         .BLKB   2
                                66  5B  1B  002E8 P.AAX:  .BYTE   27, 91, 102                         ;
                                            002EB         .BLKB   1
                                    48  1B  002EC P.AAY:  .BYTE   27, 72                              ;
                                            002EE         .BLKB   2
                                    41  1B  002F0 P.AAZ:  .ASCII  <27>\A\                             ;
                                    41  1B  002F2         .ASCII  <27>\A\                             ;
                                    41  1B  002F4         .ASCII  <27>\A\                             ;
```

```
                                        41   1B  002F6              .ASCII   <27>\A\                                    ;
                                        41   1B  002F8              .ASCII   <27>\A\                                    ;
                                        41   1B  002FA              .ASCII   <27>\A\                                    ;
                                        41   1B  002FC              .ASCII   <27>\A\                                    ;
                                        41   1B  002FE              .ASCII   <27>\A\                                    ;
                                        41   1B  00300              .ASCII   <27>\A\                                    ;
                                        41   1B  00302              .ASCII   <27>\A\                                    ;
                                        41   1B  00304              .ASCII   <27>\A\                                    ;
                                        41   1B  00306              .ASCII   <27>\A\                                    ;
                                        41   1B  00308              .ASCII   <27>\A\                                    ;
                                        41   1B  0030A              .ASCII   <27>\A\                                    ;
                                        41   1B  0030C              .ASCII   <27>\A\                                    ;
                                        41   1B  0030E              .ASCII   <27>\A\                                    ;
                                        41   1B  00310              .ASCII   <27>\A\                                    ;
                                        41   1B  00312              .ASCII   <27>\A\                                    ;
                                        41   1B  00314              .ASCII   <27>\A\                                    ;
                                        41   1B  00316              .ASCII   <27>\A\                                    ;
                                        41   1B  00318              .ASCII   <27>\A\                                    ;
                                        41   1B  0031A              .ASCII   <27>\A\                                    ;
                                        41   1B  0031C              .ASCII   <27>\A\                                    ;
                                        41   1B  0031E              .ASCII   <27>\A\                                    ;
                                        42   1B  00320  P.ABA:      .BYTE    27, 66                                     ;
                                                 00322              .BLKB    2
                                        43   1B  00324  P.ABB:      .BYTE    27, 67                                     ;

                                                        UP=                        65
                                                        DOWN=                      66
                                                        RIGHT=                     67
                                                        UP_24=                     P.AAZ


                                           0FFC  00000              .ENTRY   COB$$SET_CURSOR_REL, Save R2,R3,R4,R5,R6,-    ; 2552
                                                                             R7,R8,R9,R10,R11                             ;
                     5B  00000000G      00   9E  00002              MOVAB    SYS$FAOL, R11                                ;
                     5A         9E      AF   9E  00009              MOVAB    P.AAU, R10                                   ;
                     5E                 38   C2  0000D              SUBL2    #56, SP                                      ;
              30     AE  010E0006       8F   D0  00010              MOVL     #17694726, UP_CTL                            ; 2645
              34     AE                 6A   9E  00018              MOVAB    P.AAU, UP_CTL+4                              ;
              28     AE  010E0006       8F   D0  0001C              MOVL     #17694726, DOWN_CTL                          ;
              2C     AE         08      AA   9E  00024              MOVAB    P.AAV, DOWN_CTL+4                            ;
              20     AE  010E0006       8F   D0  00029              MOVL     #17694726, RIGHT_CTL                         ;
              24     AE         10      AA   9E  00031              MOVAB    P.AAW, RIGHT_CTL+4                           ;
                     59         04      AC   D0  00036              MOVL     TERM_TYPE, R9                                ; 2732
                     03                 59   D1  0003A              CMPL     R9, #3                                       ;
                                        08   13  0003D              BEQL     1$                                           ;
                     02                 59   D1  0003F              CMPL     R9, #2                                       ; 2733
                                        03   13  00042              BEQL     1$                                           ;
                                      0181   31  00044              BRW      23$                                          ;
                     56         1C      AC   D0  00047  1$:         MOVL     CUR_SIZE, R6                                 ; 2737
       55     18     AC                 66   C1  0004B              ADDL3    (R6), BUFFER, FREE_ADDR                      ;
                     57         08      AC   D0  00050              MOVL     LINE_NO, R7                                  ; 2739
                                        58   D4  00054              CLRL     R8                                           ;
                                        57   D5  00056              TSTL     R7                                           ;
                                        22   13  00058              BEQL     2$                                           ;
                                        58   D6  0005A              INCL     R8                                           ;
                                0C      AC   D5  0005C              TSTL     COL_NO                                       ; 2740
                                        1B   13  0005F              BEQL     2$                                           ;
```

```
                 52     0C   AC        14   AC  C1 00061            ADDL3   COL_PLUS, COL_NO, R2                      ; 2744
                             54             56  D0 00067            MOVL    R6, R4                                    ; 2743
                             53        18   AC  D0 0006A            MOVL    BUFFER, R3                                :
                             51             57  D0 0006E            MOVL    R7, R1                                    :
                             50             59  D0 00071            MOVL    R9, R0                                    :
                                          FEB1  30 00074            BSBW    COB$$SET_CURSOR_ABS_R4                    :
                 55     18   AC             66  C1 00077            ADDL3   (R6), BUFFER, FREE_ADDR                   ; 2746
                             03             58  E8 0007C  2$:       BLBS    R8, 4$                                    ; 2749
                                          00B9  31 0007F  3$:       BRW     14$                                       :
                                       0C   AC  D5 00082  4$:       TSTL    COL_NO                                    ; 2750
                                            F8  12 00085            BNEQ    3$                                        :
                                       14   AC  D5 00087            TSTL    COL_PLUS                                  ; 2753
                                            20  12 0008A            BNEQ    6$                                        :
                                            58  D4 0008C            CLRL    R8                                        ; 2756
                        03                  59  D1 0008E            CMPL    R9, #3                                    :
                                            10  12 00091            BNEQ    5$                                        :
                                            58  D6 00093            INCL    R8                                        :
      85         18          00        18   AA  F0 00095            INSV    P.AAX, #0, #24, (FREE_ADDR)+              ; 2757
                             55             02  C0 0009B            ADDL2   #2, FREE_ADDR                             :
                             66             03  C0 0009E            ADDL2   #3, (R6)                                  :
                                            4D  11 000A1            BRB     8$                                        ; 2756
                             85        1C   AA  B0 000A3  5$:       MOVW    P.AAY, (FREE_ADDR)+                       ; 2759
                             66             02  C0 000A7            ADDL2   #2, (R6)                                  :
                                            44  11 000AA            BRB     8$                                        ; 2753
                                            58  D4 000AC  6$:       CLRL    R8                                        ; 2769
                        03                  59  D1 000AE            CMPL    R9, #3                                    :
                                            32  12 000B1            BNEQ    7$                                        :
                                            58  D6 000B3            INCL    R8                                        :
                             6E             18  D0 000B5            MOVL    #24, CVT_ARG                              ; 2771
                        18   AE  010E000F   8F  D0 000B8            MOVL    #17694735, FAO_BUF                        :
                        1C   AE             55  D0 000C0            MOVL    FREE_ADDR, FAO_BUF+4                      :
                                            5E  DD 000C4            PUSHL   SP                                        :
                                       1C   AE  9F 000C6            PUSHAB  FAO_BUF                                   :
                                       0C   AE  9F 000C9            PUSHAB  FAO_LEN                                   :
                                       3C   AE  9F 000CC            PUSHAB  UP_CTL                                    :
                             6B             04  FB 000CF            CALLS   #4, SYS$FAOL                              :
                             43             50  E9 000D2            BLBC    STATUS, 9$                                :
                             50        04   AE  3C 000D5            MOVZWL  FAO_LEN, R0                               :
                             66             50  C0 000D9            ADDL2   R0, (R6)                                  :
                             50        04   AE  3C 000DC            MOVZWL  FAO_LEN, R0                               :
                             55             50  C0 000E0            ADDL2   R0, FREE_ADDR                             :
                                            0B  11 000E3            BRB     8$                                        ; 2769
                 65     20   AA             30  28 000E5  7$:       MOVC3   #48, UP_24, (FREE_ADDR)                   ; 2774
                             55             53  D0 000EA            MOVL    R3, FREE_ADDR                             :
                             66             30  C0 000ED            ADDL2   #48, (R6)                                 ; 2775
                             39             58  E9 000F0  8$:       BLBC    R8, 11$                                   ; 2781
                             01             57  D1 000F3            CMPL    R7, #1                                    ; 2783
                                            43  13 000F6            BEQL    14$                                       :
                        08   AE        FF   A7  9E 000F8            MOVAB   -1(R7), CVT_ARG                           :
                        18   AE  010E000F   8F  D0 000FD            MOVL    #17694735, FAO_BUF                        :
                        1C   AE             55  D0 00105            MOVL    FREE_ADDR, FAO_BUF+4                      :
                                       08   AE  9F 00109            PUSHAB  CVT_ARG                                   :
                                       1C   AE  9F 0010C            PUSHAB  FAO_BUF                                   :
                                       14   AE  9F 0010F            PUSHAB  FAO_LEN                                   :
                                       34   AE  9F 00112            PUSHAB  DOWN_CTL                                  :
                             6B             04  FB 00115            CALLS   #4, SYS$FAOL                              :
                             01             50  E8 00118  9$:       BLBS    STATUS, 10$                               :
                                   04 0011B         RET                                              ;
                   50        0C AE 3C 0011C 10$:    MOVZWL  FAO_LEN, R0                              ;
                   66           50 C0 00120         ADDL2   R0, (R6)                                 ;
                   50        0C AE 3C 00123         MOVZWL  FAO_LEN, R0                              ;
                   55           50 C0 00127         ADDL2   R0, FREE_ADDR                            ;
                                0F 11 0012A         BRB     14$                                      ; 2781
                                50 D4 0012C 11$:    CLRL    COUNTER                                  ; 2785
                                07 11 0012E         BRB     13$                                      ;
                   85        50 AA B0 00130 12$:    MOVW    P.ABA, (FREE_ADDR)+                      ;
                   66           02 C0 00134         ADDL2   #2, (R6)                                 ;
             F5    50           57 F2 00137 13$:    AOBLSS  R7, COUNTER, 12$                         ;
                                58 D4 0013B 14$:    CLRL    R8                                       ; 2788
                                57 D5 0013D         TSTL    R7                                       ;
                                0C 12 0013F         BNEQ    15$                                      ;
                                58 D6 00141         INCL    R8                                       ;
                             0C AC D5 00143         TSTL    COL_NO                                   ; 2789
                                05 13 00146         BEQL    15$                                      ;
                   85           0D 90 00148         MOVB    #13, (FREE_ADDR)+                        ; 2792
                                66 D6 0014B         INCL    (R6)                                     ; 2794
                   57        10 AC D0 0014D 15$:    MOVL    LINE_PLUS, R7                            ; 2797
                                0C 13 00151         BEQL    16$                                      ;
     57            0A        6E 00 2C 00153         MOVC5   #0, (SP), #10, R7, (FREE_ADDR)           ; 2800
                                65    00158                                                          ;
                   55           53 D0 00159         MOVL    R3, FREE_ADDR                            ;
                   66           57 C0 0015C         ADDL2   R7, (R6)                                 ; 2801
                             14 AC D5 0015F 16$:    TSTL    COL_PLUS                                 ; 2804
                                05 12 00162         BNEQ    17$                                      ;
                             0C AC D5 00164         TSTL    COL_NO                                   ;
                                5F 13 00167         BEQL    23$                                      ;
                   05           58 E8 00169 17$:    BLBS    R8, 18$                                  ; 2805
                             0C AC D5 0016C         TSTL    COL_NO                                   ;
                                57 12 0016F         BNEQ    23$                                      ;
             50        0C AC    01 C3 00171 18$:    SUBL3   #1, COL_NO, COL                          ; 2810
                                02 18 00176         BGEQ    19$                                      ; 2811
                                50 D4 00178         CLRL    COL                                      ; 2813
                   50        14 AC C0 0017A 19$:    ADDL2   COL_PLUS, R0                             ; 2816
                   03           59 D1 0017E         CMPL    R9, #3                                   ; 2814
                                36 12 00181         BNEQ    20$                                      ;
                                50 D5 00183         TSTL    R0                                       ; 2816
                                41 13 00185         BEQL    23$                                      ;
                10 AE           50 D0 00187         MOVL    R0, CVT_ARG                              ;
                18 AE 010E000F  8F D0 0018B         MOVL    #17694735, FAO_BUF                       ;
                1C AE           55 D0 00193         MOVL    FREE_ADDR, FAO_BUF+4                     ;
                             10 AE 9F 00197         PUSHAB  CVT_ARG                                  ;
                             1C AE 9F 0019A         PUSHAB  FAO_BUF                                  ;
                             1C AE 9F 0019D         PUSHAB  FAO_LEN                                  ;
                             2C AE 9F 001A0         PUSHAB  RIGHT_CTL                                ;
                   6B           04 FB 001A3         CALLS   #4, SYS$FAOL                             ;
                   22           50 E9 001A6         BLBC    STATUS, 24$                              ;
                   50        14 AE 3C 001A9         MOVZWL  FAO_LEN, R0                              ;
                   66           50 C0 001AD         ADDL2   R0, (R6)                                 ;
                   50        14 AE 3C 001B0         MOVZWL  FAO_LEN, R0                              ;
                   55           50 C0 001B4         ADDL2   R0, FREE_ADDR                            ;
                                0F 11 001B7         BRB     23$                                      ; 2814
                                51 D4 001B9 20$:    CLRL    COUNTER                                  ; 2818
                                07 11 001BB         BRB     22$                                      ;
                   85        54 AA B0 001BD 21$:    MOVW    P.ABB, (FREE_ADDR)+                      ;
```

```
                               66          02  C0 001C'          ADDL2   #2, (R6)                                  ;
                   F5          51          50  F3 001C4 22$:     AOBLEQ  R0, COUNTER, 21$                          ;
                               50          01  D0 001C8 23$:     MOVL    #1, R0                                    ; 2821
                                               04 001CB 24$:     RET                                               ; 2823

; Routine Size:  460 bytes,    Routine Base:  _COB$CODE + 0326


; 1247          2824  1 !<BLF/PAGE>
```

```
%SBTTL 'COB$$SETUP_TERM_TYPE - Setup terminal type for COB$$ routines'
GLOBAL ROUTINE COB$$SETUP_TERM_TYPE (
                     FILE_NAME,
                     NAME_LEN,
                     TERM_TYPE,
                     SEC_DEV_CHAR,
                     DEVICE_TYPE  : REF VECTOR [,BYTE],
                     RES_NAME_LEN : REF VECTOR [,WORD],
                     RES_NAME_ADDR
                 ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine uses the specified file name to determine device
!       characteristics and assign a terminal type code which is understood
!       by other COB$$ routines.  COB$$ routines use the terminal type to
!       determine the correct escape sequence for a given function (ex. set
!       cursor).
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = COB$$SETUP_TERM_TYPE (FILE_NAME.rt.r,
!                                                NAME_LEN.rl.v,
!                                                TERM_TYPE.wl.r
!                                                [,SEC_DEV_CHAR.wlu.r]
!                                                [,DEVICE_TYPE.wbu.r]
!                                                [,RES_NAME_LEN.wwu.r,
!                                                  RES_NAME_ADDR.wt.r])
!
! FORMAL PARAMETERS:
!
!       FILE_NAME.rt.r          addr of file name text
!       NAME_LEN.rl.v           length of file name text
!       TERM_TYPE.wl.r          terminal type code, one of the following:
!                                   unknown
!                                   vt05
!                                   vt52
!                                   vt100
!                                   vtforeign
!                                   hardcopy
!
!       SEC_DEV_CHAR.wlu.r      [Optional] If supplied, the address of
!                               a longword to receive the secondary
!                               device dependent bits.  This is the
!                               field that, e.g. tells whether a VT100
!                               has AVO.
!
!       DEVICE_TYPE.wbu.r       [Optional].  If present, address of byte
!                               to receive hardware device type.  These
!                               are the DT$_type codes.
!
!       RES_NAME_LEN.wwu.r      [Optional -- if provided, RES_NAME_ADDR
!                               must be provided as well.]  If present,
!                               the address of a word to receive the
!                               length of the resultant name string.
!
!       RES_NAME_ADDR.wt.r      [Optional -- if provided, RES_NAME_LEN
1 !                                         must be provided as well.]  If present,
1 !                                         the address of a buffer to receive the
1 !                                         resultant name string.  NOTE:  This
1 !                                         routine assumes that the supplied buffer
1 !                                         is large enough to contain the resultant
1 !                                         name string.  It must be a minimum of 4
1 !                                         bytes long and should be at least 64
1 !                                         bytes long to guarantee that the name
1 !                                         will fit.
1 ! IMPLICIT INPUTS:
1 !
1 !      NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !      NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !      NONE
1 !--
1
2    BEGIN
2
2    BUILTIN
2        NULLPARAMETER;
2
2    LOCAL
2        DEVNAM_DSC : BLOCK [8, BYTE],            ! dsc for name
2        DVI_ITMLST : VECTOR [3*3 + 1] INITIAL    ! item list for $GETDVI
2                (DVI$_DEVTYPE    ^ 16 + 4, 0, 0,!  device type
2                 DVI$_DEVDEPEND2 ^ 16 + 4, 0, 0,!  device dependent bits
2                 DVI$_DEVNAM     ^ 16 +64, 0, 0,!  result name string
2                 0),                            !  terminater
2
2        DVI_EFN,                                 ! event flag for $GETDVI,
2        STATUS,                                  ! status retd by called routines
2        DEV_TYPE : VOLATILE,                     ! storage for $GETDVI value
2        DEV_DEPEND2 : VOLATILE,                  ! storage for $GETDVI value
2
2        DEV_DEVNAM : VECTOR [64, BYTE],          ! Buffer for result name
2                                                 ! string
2
2        DEV_NAMLEN : VOLATILE WORD;              ! Length of returned
2                                                 ! resultent name string
2    BIND
2        DVI_TYPE    = DVI_ITMLST + 4,            ! make it easy to reference
2        DVI_DEPEND2 = DVI_ITMLST + 16,           !  items retd by $GETDVI
2        DVI_DEVNAM  = DVI_ITMLST + 28,           !
2        DVI_NAMLEN  = DVI_ITMLST + 32;           !
2
2    MAP
2        DEV_DEPEND2 : BLOCK [4, BYTE];
2
```

```
  2      DVI_TYPE    = DEV_TYPE;                        ! fill in rest of itmlst
  2      DVI_DEPEND2 = DEV_DEPEND2;
  2      DVI_DEVNAM  = DEV_DEVNAM;
  2      DVI_NAMLEN  = DEV_NAMLEN;
  2
  3      IF NOT (STATUS = LIB$GET_EF (DVI_EFN))
  2      THEN RETURN (.STATUS);                 ! get unique event flag number
  2
  2      DEVNAM_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
  2      DEVNAM_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
  2      DEVNAM_DSC [DSC$W_LENGTH] = .NAME_LEN;
  2      DEVNAM_DSC [DSC$A_POINTER] = .FILE_NAME; ! dsc needed for $GETDVI
  2
P 2      STATUS = $GETDVI (EFN = .DVI_EFN, DEVNAM = DEVNAM_DSC,
  2                        ITMLST = DVI_ITMLST);
  2      IF NOT .STATUS THEN RETURN (.STATUS);
  2
  2      $WAITFR (EFN = .DVI_EFN);              ! make $GETDVI synchronous
  2
  3      IF NOT (STATUS = LIB$FREE_EF (DVI_EFN))
  2      THEN RETURN (.STATUS);                 ! free event flag
  2
  2      SELECTONE .DEV_TYPE OF
  2      SET
  2          [DT$_VT100]:
  2              .TERM_TYPE = VT100;
  2
  2          [DT$_VT52, DT$_VT55]:
  2              .TERM_TYPE = VT52;
  2
  2          [DT$_VT05]:
  2              .TERM_TYPE = VT05;
  2
  2          [DT$_FT1 TO DT$_FT2]:
  2              .TERM_TYPE = VTFOREIGN;
  2
  2          [DT$_LA36, DT$_LA120, DT$_LA34, DT$_LA38]:
  2              .TERM_TYPE = HARDCOPY;
  2
  2          [OTHERWISE]:
  2              IF .DEV_DEPEND2 [TT2$V_DECCRT] OR
  2                 .DEV_DEPEND2 [TT2$V_ANSICRT]
  2              THEN
  2                  .TERM_TYPE = VT100       ! VT100 compatible (ANSI)
  2              ELSE
  2                  .TERM_TYPE = UNKNOWN;    ! really unknown
  2          TES;
  2
  2  !+
  2  ! Return optional parameters if requested.
  2  !-
  2      IF NOT NULLPARAMETER (4)
  2      THEN
  2          .SEC_DEV_CHAR = .DEV_DEPEND2;
  2
  2      IF NOT NULLPARAMETER (5)
  2      THEN
```

```
; 1420          2996 2          DEVICE_TYPE [0] = .DEV_TYPE;
; 1421          2997 2
; 1422          2998 2      IF NOT NULLPARAMETER (6)  AND
; 1423          2999 2         NOT NULLPARAMETER (7)
; 1424          3000 2      THEN
; 1425          3001 3          BEGIN
; 1426          3002 3          CH$MOVE ( .DEV_NAMLEN, DEV_DEVNAM, .RES_NAME_ADDR);
; 1427          3003 3          RES_NAME_LEN [0] = .DEV_NAMLEN;
; 1428          3004 2          END;
; 1429          3005 2
; 1430          3006 2      RETURN (.STATUS);
; 1431          3007 1      END;                                ! End of routine COB$$SETUP_TERM_TYPE


                                                                     004F2          .BLKB    2
00000000  00000000  001C0004  00000000  00000000  00060004  004F4 P.ABC:  .LONG    393220, 0, 0, 1835012, 0, 0, 2097216, 0, -  ;
                    00000000  00000000  00000000  00200040  0050C                   0, 0                                          ;

                                                                                    .EXTRN   SYS$GETDVI, SYS$WAITFR

                                                              007C 00000            .ENTRY   COB$$SETUP_TERM_TYPE, Save R2,R3,R4,R5,R6   ; 2826
                              5E          80     AE  9E 00002            MOVAB    -128(SP), SP                                ;
                    50   AE   CE   AF            28  28 00006            MOVC3    #40, P.ABC, DVI_ITMLST                       ; 2918
                              54   AE     4C     AE  9E 0000C            MOVAB    DEV_TYPE, DVI_TYPE                           ; 2939
                              60   AE     48     AE  9E 00011            MOVAB    DEV_DEPEND2, DVI_DEPEND2                     ; 2940
                              6C   AE     08     AE  9E 00016            MOVAB    DEV_DEVNAM, DVI_DEVNAM                       ; 2941
                              70   AE     06     AE  9E 0001B            MOVAB    DEV_NAMLEN, DVI_NAMLEN                       ; 2942
                                                 5E  DD 00020            PUSHL    SP                                           ; 2944
                       00000000G   00            01  FB 00022            CALLS    #1, LIB$GET_EF                               ;
                                   56            50  D0 00029            MOVL     R0, STATUS                                   ;
                                   41            56  E9 0002C            BLBC     STATUS, 1$                                   ;
                              7A   AE   010E     8F  B0 0002F            MOVW     #270, DEVNAM_DSC+2                           ; 2947
                              78   AE     08     AC  B0 00035            MOVW     NAME_LEN, DEVNAM_DSC                         ; 2949
                              7C   AE     04     AC  D0 0003A            MOVL     FILE_NAME, DEVNAM_DSC+4                      ; 2950
                                                 7E  7C 0003F            CLRQ     -(SP)                                        ; 2953
                                                 7E  7C 00041            CLRQ     -(SP)                                        ;
                                          60     AE  9F 00043            PUSHAB   DVI_ITMLST                                   ;
                                          F8     AD  9F 00046            PUSHAB   DEVNAM_DSC                                   ;
                                                 7E  D4 00049            CLRL     -(SP)                                        ;
                                          1C     AE  DD 0004B            PUSHL    DVI_EFN                                      ;
                       00000000G   00            08  FB 0004E            CALLS    #8, SYS$GETDVI                               ;
                                   56            50  D0 00055            MOVL     R0, STATUS                                   ;
                                   15            56  E9 00058            BLBC     STATUS, 1$                                   ; 2954
                                                 6E  DD 0005B            PUSHL    DVI_EFN                                      ; 2956
                       00000000G   00            01  FB 0005D            CALLS    #1, SYS$WAITFR                               ;
                                                 5E  DD 00064            PUSHL    SP                                           ; 2958
                       00000000G   00            01  FB 00066            CALLS    #1, LIB$FREE_EF                              ;
                                   56            50  D0 0006D            MOVL     R0, STATUS                                   ;
                                   03            56  E8 00070 1$:        BLBS     STATUS, 2$                                   ;
                                               009C  31 00073            BRW      12$                                          ;
                                   50     4C     AE  D0 00076 2$:        MOVL     DEV_TYPE, R0                                 ; 2961
                       00000060    8F            50  D1 0007A            CMPL     R0, #96                                      ; 2963
                                   48            13 00081                BEQL     7$                                           ;
                                   3F            50  D1 00083            CMPL     R0, #63                                      ; 2966
                                   0F            15 00086                BLEQ     3$                                           ;
                       00000041    8F            50  D1 00088            CMPL     R0, #65                                      ;
```

```
                                          06  14 0008F          BGTR    3$                                        ;
                             0C   BC      02  D0 00091          MOVL    #2, @TERM_TYPE                            ; 2967
                                          3D  11 00095          BRB     9$                                        ;
                                  01      50  D1 00097 3$:      CMPL    R0, #1                                    ; 2969
                                          06  12 0009A          BNEQ    4$                                        ;
                             0C   BC      01  D0 0009C          MOVL    #1, @TERM_TYPE                            ; 2970
                                          32  11 000A0          BRB     9$                                        ;
                                  10      50  D1 000A2 4$:      CMPL    R0, #16                                   ; 2972
                                          0B  19 000A5          BLSS    5$                                        ;
                                  11      50  D1 000A7          CMPL    R0, #17                                   ;
                                          06  14 000AA          BGTR    5$                                        ;
                             0C   BC      04  D0 000AC          MOVL    #4, @TERM_TYPE                            ; 2973
                                          22  11 000B0          BRB     9$                                        ;
                                  20      50  D1 000B2 5$:      CMPL    R0, #32                                   ; 2975
                                          0B  19 000B5          BLSS    6$                                        ;
                                  23      50  D1 000B7          CMPL    R0, #35                                   ;
                                          06  14 000BA          BGTR    6$                                        ;
                             0C   BC      05  D0 000BC          MOVL    #5, @TERM_TYPE                            ; 2976
                                          12  11 000C0          BRB     9$                                        ;
                   04        4B   AE      05  E0 000C2 6$:      BBS     #5, DEV_DEPEND2+3, 7$                     ; 2979
                                  06  4B  AE  E9 000C7          BLBC    DEV_DEPEND2+3, 8$                         ; 2980
                             0C   BC      03  D0 000CB 7$:      MOVL    #3, @TERM_TYPE                            ; 2982
                                          03  11 000CF          BRB     9$                                        ;
                                      0C  BC  D4 000D1 8$:      CLRL    @TERM_TYPE                                ; 2984
                                  04      6C  91 000D4 9$:      CMPB    (AP), #4                                  ; 2990
                                          0A  1F 000D7          BLSSU   10$                                       ;
                                      10  AC  D5 000D9          TSTL    16(AP)                                    ;
                                          05  13 000DC          BEQL    10$                                       ;
                             10   BC  48  AE  D0 000DE          MOVL    DEV_DEPEND2, @SEC_DEV_CHAR                ; 2992
                                  05      6C  91 000E3 10$:     CMPB    (AP), #5                                  ; 2994
                                          0A  1F 000E6          BLSSU   11$                                       ;
                                      14  AC  D5 000E8          TSTL    20(AP)                                    ;
                                          05  13 000EB          BEQL    11$                                       ;
                             14   BC  4C  AE  90 000ED          MOVB    DEV_TYPE, @DEVICE_TYPE                    ; 2996
                                  06      6C  91 000F2 11$:     CMPB    (AP), #6                                  ; 2998
                                          1B  1F 000F5          BLSSU   12$                                       ;
                                      18  AC  D5 000F7          TSTL    24(AP)                                    ;
                                          16  13 000FA          BEQL    12$                                       ;
                                  07      6C  91 000FC          CMPB    (AP), #7                                  ; 2999
                                          11  1F 000FF          BLSSU   12$                                       ;
                                      1C  AC  D5 00101          TSTL    28(AP)                                    ;
                                          0C  13 00104          BEQL    12$                                       ;
              1C    BC       08   AE  06  AE  28 00106          MOVC3   DEV_NAMLEN, DEV_DEVNAM, @RES_NAME_ADDR    ; 3002
                             18   BC  06  AE  B0 0010D          MOVW    DEV_NAMLEN, @RES_NAME_LEN                 ; 3003
                                  50      56  D0 00112 12$:     MOVL    STATUS, R0                                ; 3006
                                              04 00115          RET                                               ; 3007

; Routine Size:  278 bytes,    Routine Base:  _COB$CODE + 051C


; 1432          3008  1 !<BLF/PAGE>
```

```
1 %SBTTL 'COB$$UP_SCROLL_R2 - Create up scroll sequence'
1 GLOBAL ROUTINE COB$$UP_SCROLL_R2 (
1                       TERM_TYPE,
1                       BUFFER,
1                       CUR_SIZE
1                   ) : COB$$ESC_R2_LNK =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine generates the escape sequence for up scroll.
1 !       The string is appended into the buffer.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = COB$$UP_SCROLL_R2 (TERM_TYPE.rl.v, BUFFER.mt.r,
1 !                                             CUR_SIZE.ml.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       TERM_TYPE.rl.v          terminal type
1 !       BUFFER.mt.r             addr of buffer
1 !       CUR_SIZE.ml.r           # bytes currently in buffer
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     LOCAL
2         FREE_ADDR : REF VECTOR [,BYTE];
2
2     FREE_ADDR = .BUFFER + ..CUR_SIZE;
2
2     CASE .TERM_TYPE FROM UNKNOWN TO HARDCOPY OF
2     SET
2         [VT05]:
3             BEGIN
3             FREE_ADDR [0] = LF;
3             FREE_ADDR [1] = NULL;
3             FREE_ADDR [2] = NULL;
3             FREE_ADDR [3] = NULL;
3             .CUR_SIZE = ..CUR_SIZE + 4;
2             END;
2
```

```
; 1491          3066  2           [VT52, VT100]:
; 1492          3067  3               BEGIN
; 1493          3068  3               FREE_ADDR [0] = LF;
; 1494          3069  3               .CUR_SIZE = ..CUR_SIZE + 1;
; 1495          3070  2               END;
; 1496          3071  2
; 1497          3072  2           [HARDCOPY, UNKNOWN, VTFOREIGN]:
; 1498          3073  2               ;
; 1499          3074  2
; 1500          3075  2           [INRANGE, OUTRANGE]:
; 1501          3076  2               RETURN 0;                     ! should never get here
; 1502          3077  2
; 1503          3078  2           TES;
; 1504          3079  2
; 1505          3080  2       RETURN (SS$_NORMAL);
; 1506          3081  2
; 1507          3082  1       END;                              ! End of routine COB$$UP_SCROLL_R2



                                    51          62  C0 00000 COB$$UP_SCROLL_R2::
                                                                    ADDL2   (CUR_SIZE), FREE_ADDR                ; 3053
                             05     00          50  CF 00003        CASEL   TERM_TYPE, #0, #5                    ; 3055
         0016              0016   000E        001B     00007 1$:    .WORD   4$-1$,-                              ;
                                  001B        001B     0000F                2$-1$,-                              ;
                                                                            3$-1$,-                              ;
                                                                            3$-1$,-                              ;
                                                                            4$-1$,-                              ;
                                                                            4$-1$                                ;
                                                11  11 00013        BRB     5$                                   ; 3076
                                    61          0A  D0 00015 2$:    MOVL    #10, (FREE_ADDR)                     ; 3059
                                    62          04  C0 00018        ADDL2   #4, (CUR_SIZE)                       ; 3063
                                                05  11 0001B        BRB     4$                                   ; 3055
                                    61          0A  90 0001D 3$:    MOVB    #10, (FREE_ADDR)                     ; 3068
                                                62  D6 00020        INCL    (CUR_SIZE)                           ; 3069
                                    50          01  D0 00022 4$:    MOVL    #1, R0                               ; 3080
                                                    05 00025        RSB                                          ;
                                                50  D4 00026 5$:    CLRL    R0                                   ; 3082
                                                    05 00028        RSB                                          ;

; Routine Size:  41 bytes,    Routine Base:  _COB$CODE + 0632

; 1508          3083  1 !<BLF/PAGE>
; 1510          3084  1 END                                             . End of module COB$$ESCAPE_GENERATOR
; 1511          3085  1
; 1512          3086  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                        Bytes                           Attributes
;
;  _COB$CODE                            1627  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       30         0         581         00:00.7
;  _$255$DUA28:[COBRTL.OBJ]SMGLIB.L32;1           469       31         6          38         00:00.2



;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:COBESCGEN/OBJ=OBJ$:COBESCGEN MSRC$:COBESCGEN/UPDATE=(ENH$:COBESCGEN
;       )
;
; Size:         1396 code + 231 data bytes
; Run Time:        00:24.7
; Elapsed Time:    01:33.0
; Lines/CPU Min:     7484
; Lexemes/CPU-Min: 27092
; Memory Used:  234 pages
; Compilation Complete
```

0062 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

COBDIVQ
LIS

COBFINDNA
LIS

COBDBEXCE
LIS

COBEXPI
LIS

COBDEEDIT
LIS

COBDISPLA
LIS

COBESCGEN
LIS

COBERROR
LIS

COBDHANDL
LIS