

```

CBF2:A5 21      28      LDA  WNDWDTH  ;get width of screen window
CBF4:48          29      PHA           ;save original width
CBF5:2C 1F CO   30      BIT   RD8OVID  ;in 40 or 80 columns?
CBF8:10 1C CC16 31      BPL   GETST1   ;=>40, determine starting line
CBFA:8D 01 CO   32      STA   SET8OCOL ;make sure this is enabled
CBFD:4A          33      LSR   A       ;divide by 2 for 80 column index
CBFE:AA          34      TAX           ;and save
CBFF:A5 20      35      LDA  WNDLFT   ;test oddity of right edge
CC01:4A          36      LSR   A       ;by rotating low bit into carry
CC02:88          37      CLV           ;V=0 if left edge even
CC03:90 03 CC08 38      BCC   CHKRT   ;=>check right edge
CC05:2C 06 CB   39      BIT   SEV1    ;V=1 if left edge odd
CC08:2A          40      CHKRT  ROL   A ;restore WNDLFT
CC09:45 21      41      EOR   WNDWDTH ;get oddity of right edge
CC0B:4A          42      LSR   A       ;C=1 if right edge even
CC0C:70 03 CC11 43      BVS   GETST   ;if odd left, don't DEY
CC0E:B0 01 CC11 44      BCS   GETST   ;if even right, don't DEY
CC10:CA          45      DEX           ;if right edge odd, need one less
CC11:86 21      46      GETST  STX   WNDWDTH ;save window width
CC13:AD 1F CO   47      LDA  RD8OVID  ;N=1 if 80 columns
CC16:08          48      GETST1 PHP    ;save N,Z,V
CC17:A6 22      49      LDX   WNDTOP   ;assume scroll from top
CC19:98          50      TYA           ;up or down?
CC1A:D0 03 CC1F 51      BNE   SETDBAS  ;=>up
CC1C:A6 23      52      LDX   WNDBTM   ;down, start scrolling at bottom
CC1E:CA          53      DEX           ;really need one less
CC1F:          54      *
CC1F:8A          55      SETDBAS TXA   ;get current line
CC20:20 03 CE   56      JSR   VTABZ    ;calculate base with window width
CC23:          57      *
CC23:A5 28      58      SCRLIN  LDA   BASL   ;current line is destination
CC25:85 2A      59      STA   BAS2L
CC27:A5 29      60      LDA   BASH
CC29:85 2B      61      STA   BAS2H
CC2B:          62      *
CC2B:AD 7B 07   63      LDA  TEMP1    ;test direction
CC2E:F0 32 CC62 64      BEQ   SCRLDN  ;=>do the downer
CC30:E8          65      INX           ;do next line
CC31:E4 23      66      CPX   WNDBTM   ;done yet?
CC33:B0 32 CC67 67      BCS   SCRL3   ;=>yup, all done
CC35:8A          68      SETSRC  TXA   ;set new line
CC36:20 03 CE   69      JSR   VTABZ    ;get base for new current line
CC39:A4 21      70      LDY   WNDWDTH  ;get width for scroll
CC3B:2B          71      PLP           ;get status for scroll
CC3C:08          72      PHP           ;N=1 if 80 columns
CC3D:10 1E CC5D 73      BPL   SKPRT   ;=>only do 40 columns
CC3F:AD 55 CO   74      LDA  TXTPAGE2 ;scroll aux page first (even bytes)
CC42:98          75      TYA           ;test Y
CC43:F0 07 CC4C 76      BEQ   SCRLFT   ;if Y=0, only scroll one byte
CC45:B1 28      77      SCRLVEN LDA  (BASL),Y
CC47:91 2A      78      STA  (BAS2L),Y
CC49:88          79      DEY           ;
CC4A:D0 F9 CC45 80      BNE   SCRLVEN  ;do all but last even byte
CC4C:70 04 CC52 81      SCRLFT  BVS   SKPLFT ;odd left edge, skip this byte

```

```

CC4E:B1 28      82      LDA  (BASL),Y
CC50:91 2A      83      STA  (BAS2L),Y
CC52:AD 54 CO   84      SKPLFT  LDA  TXTPAGE1 ;now do main page (odd bytes)
CC55:A4 21      85      LDY   WNDWDTH  ;restore width
CC57:B0 04 CC5D 86      BCS   SKPRT   ;even right edge, skip this byte
CC59:B1 28      87      SCRLDD  LDA  (BASL),Y
CC5B:91 2A      88      STA  (BAS2L),Y
CC5D:88          89      SKPRT  DEY
CC5E:10 F9 CC59 90      BPL   SCRLDD
CC60:30 C1 CC23 91      BMI   SCRLIN  ;=> always scroll next line
CC62:          92      *
CC62:CA          93      SCRLDN  DEX           ;do next line
CC63:E4 22      94      CPX   WNDTOP   ;done yet
CC65:10 CE CC35 95      BPL   SETSRC  ;=>nope, not yet
CC67:          96      *
CC67:28          97      SCRL3   PLP           ;pull status off stack
CC68:68          98      PLA           ;restore window width
CC69:85 21      99      STA  WNDWDTH
CC6B:20 96 CC   100     JSR   X.SUB    ;clear current line
CC6E:20 FE CD   101     JSR   VTAB    ;restore original cursor line
CC71:68          102     PLA           ;and X
CC72:AA          103     TAX
CC73:60          104     RTS           ;done!!!
CC74:          105     *
CC74:          106     * EXECUTE CLR TO EOS:
CC74:          107     *
CC74:20 9A CC   108     X.VT   JSR   X.GS    ;CLEAR TO EOL
CC77:A5 25      109     LDA  CV       ;SAVE CV
CC79:48          110     PHA
CC7A:10 06 CC82 111     BPL   X.VTNEXT ;DO NEXT LINE (ALWAYS TAKEN)
CC7C:20 03 CE   112     X.VTLOOP JSR  VTABZ    ;set base address
CC7F:20 96 CC   113     JSR   X.SUB    ;CLEAR LINE
CC82:E6 25      114     X.VTNEXT INC  CV
CC84:A5 25      115     LDA  CV
CC86:C5 23      116     CMP   WNDBTM   ;OFF SCREEN?
CC88:90 F2 CC7C 117     BCC   X.VTLOOP ;=>NO, KEEP GOING
CC8A:68          118     PLA           ;RESTORE
CC8B:85 25      119     STA  CV       ; CV
CC8D:4C FE CD   120     JMP   VTAB    ;return via VTAB (blech)
CC90:          121     *
CC90:          122     * EXECUTE CLEAR:
CC90:          123     *
CC90:          124     X.FF   EQU   *
CC90:20 5F CB   125     JSR   X.EM     ;HOME THE CURSOR
CC93:4C 74 CC   126     JMP   X.VT    ;RETURN VIA CLREOS (UGH!!)
CC96:          127     *
CC96:          128     * EXECUTE CLEAR LINE
CC96:          129     *
CC96:A0 00      130     X.SUB  LDY   #0      ;start at left
CC98:F0 03 CC9D 131     BEQ   X.GSEOLZ ;and clear to end of line
CC9A:          132     *
CC9A:          133     * EXECUTE CLEAR TO EOL:
CC9A:          134     *
CC9A:AC 7B 05   135     X.GS   LDY   OURCH  ;get CH

```