ANSI Console Escape Sequences

To access the extended console functions of the ANSI.SYS driver, first be sure that the command

```
DEVICE=ansi.sys
```

is in your CONFIG.SYS file. Then simply send specially-formatted text sequences to the console (CON). A program may use any of the DOS Character I/O functions which display characters on the console. From the DOS command interpreter, you can TYPE a file, ECHO text sequences, or include them in your PROMPT.

In the following list, items in angle brackets «» are parameters (usually numeric values in ASCII) and the '^[' character is an ESC (ASCII 1bH). The bottom of this page shows several examples.

Sequence (^[is ESC)	Function
Cursor Motion	
^[[«row»;«clm»H	positions the cursor. Default is 1;1 (top left corner)
^[[«row»;«clm»f	(same as above)
^[[«rows»A	moves the cursor up. Default: 1. Won't go above top.
^[[«rows»B	moves cursor down. Default: 1. Won't go below bottom.
^[[«clms»C	moves the cursor forward (to the right) Default: 1
^[[«clms»D	moves the cursor backward (to the left) Default: 1
Erase Operations	
^[[2J	erases the screen and homes the cursor
^[[K	erases to the end of the current line
Miscellaneous	
^[[6n	outputs the current line and column in the form: ^[[«row»;«clm»R
^[[s	saves the current cursor position (see ^[[u)
^[[u	restores cursor to last position saved by ^[[s
Screen Control	
^[[«attr»;...;«attr»m	sets display attributes. «attr» values are: 0 normal (white on black) 1 bold (character colors are high-intensity) 4 underline (IBM Monochrome monitor only) 5 blink (foreground blinks) 7 reverse (black on white) 8 no display (foreground = background) 30 BLACK foreground 40 BLACK background 31 RED foreground 41 RED background 32 GREEN foreground 42 GREEN background 33 YELLOW foreground 43 YELLOW background 34 BLUE foreground 44 BLUE background 35 MAGENTA foreground 45 MAGENTA background 36 CYAN foreground 46 CYAN background 37 WHITE foreground 47 WHITE background

```
^[[=«mode»h              sets screen width and mode where «mode» values are:
                           0  40x25 text mode black and white
                           1  40x25 text mode color
                           2  80x25 text mode black and white
                           3  80x25 text mode color
                           4  320x200 graphics mode color
                           5  320x200 graphics mode black and white
                           6  640x200 graphics mode black and white
                           7  Causes cursor to wrap to new line at end of line.

^[[=7l                   stops cursor from wrapping at end of line

_____________________ Keyboard Redefinition _____________________________
^[[«num»;«num»...«num»p  redefines a keystroke so it yields different values.
  or                    The first «num» (or first character of «string») is
^["«string»"p            the key being redefined.  The following values are the
  or                    new value for the key.
^[[«num»;"«string»"p     The following redefines Ctrl-D key to be: DIR C:[Enter]
 or                       ^[[4;"DIR C:";13p
various
combinations            Certain keystrokes must be defined with two «num»s.
                        For example, [F1] is:  0;59; and [Home] is:  0;71;
                        The following redefines the F10 key to be: DIR [Enter]
                          ^[[0;68;"DIR";13p

                        See Extended ASCII Keystrokes for a full listing.

                        Note: To reset a key to its original value, use its
                              «num» code(s) twice. The following resets [F10].
                                   ^[[0;68;0;68p
_____________________   _____________________________________________________

To find if ANSI.SYS is installed, display the ^[[6n function and immediately read
standard input.  It should yield a value in the form:  ^[[«row»;«clm»R

------------------------------------------------------------------------------
Here are some examples usages in two programming languages:

;============================ ASM example ========================
data_seg  segment
ColorMode db  1bH,'=3h','$'
SetF10    db  1bH,'0;68;"F10 was pressed"p','$'
data_seg  ends

code_seg segment
         mov  dx,offset ColorMode  ;set screen to 80x25 color mode
         mov  ah,9                 ;DOS display string function
         int  21H
         mov  dx,offset SetF10     ;redefine the F10 key to a string of text
         mov  ah,9
         int  21H
code_seg ends

/* ====================== C function example ====================== */
set_cursor(row,clm)
int row,clm;
{
   printf("%c[%d;%dH",27,row,clm)
}
------------------------------------------------------------------------------
```

It is difficult to experiment with ANSI sequences from the DOS command prompt since DOS does not allow direct keyboard input of the ESC character. Here are three ways to experiment:

- **use the PROMPT command to output an ESC to the console. For instance:**

```
PROMPT $e[7m              sets up to display in reverse video
PROMPT $e[0;68;"DIR";13p  redefines the F10 key
```

- **use the PROMPT command to output an ESC to the console. For instance:**

```
PROMPT $e[7m              sets up to display in reverse video
PROMPT $e[0;68;"DIR";13p  redefines the F10 key
PROMPT                    sets the prompt back to normal
```

- **Create a disk file with BASIC and then TYPE it from DOS:**

```
BASIC
OPEN "ansitest.txt" FOR OUTPUT AS #1
PRINT #1,CHR$(27);"[7m"             '** display reverse video
PRINT #1,CHR$(27);"[0;68;'DIR';13p" '** redefine F10 key as DIR<Enter>
SYSTEM
TYPE ansitest.txt
```

- **Go into BASIC, open the "CON" file and use PRINT # to send escape sequences the the DOS console driver.**