VT100 ESC sequences

Control Codes

enter = '^m';	# Return
erase = '^?';	# DEL
down = '\E[B' '\EOB';	# Down Arrow
right = '\E[C' '\EOC';	# Right Arrow
up = '\E[A' '\EOA';	# Up Arrow
left = '\E[D' '\EOD' '^h';	# Left Arrow or backspace
$home = '\E[H' \mid '^aa';$	# Home or CTRL-a
nl = '\E[4~' '^j';	# Select or Linefeed
insrt = '\E[2~' '\EOR';	# Insert Here or PF3
delete = '\E[3~' '\EOS';	# Remove or PF4
eeof = '\E[1~' '\EOP';	# Find or PF1
tab = '^i';	# Tab
btab = '\E[5~';	# Prev Screen
clear = '\E[6~' '\EOQ' '^l';	# Next Screen or PF2 or CTRL-l
pfk1 = '\EOq' '\E1';	# Numeric 1 or ESC 1
pfk2 = '\EOr' '\E2';	# Numeric 2 or ESC 2
pfk3 = '\EOs' '\E3';	# Numeric 3 or ESC 3
pfk4 = '\EOt' '\E4';	# Numeric 4 or ESC 4
pfk5 = '\EOu' '\E5';	# Numeric 5 or ESC 5
pfk6 = '\EOv' '\E6';	# Numeric 6 or ESC 6
pfk7 = '\EOw' '\E7';	# Numeric 7 or ESC 7
pfk8 = '\EOx' '\E8';	# Numeric 8 or ESC 8
pfk9 = '\EOy' '\E9';	# Numeric 9 or ESC 9
pfk10 = '\EOp' '\E0';	# Numeric 0 or ESC 0
pfk11 = '\EOn' '\E!';	# Numeric Period or ESC!
pfk12 = '\EOM' '\E@';	# Enter or ESC @
pfk13 = '\E[17~' '\E#';	# F6 or ESC #
pfk14 = '\E[18~' '\E\$';	# F7 or ESC \$
pfk15 = '\E[19~' '\E%';	# F8 or ESC %
$pfk16 = \E[20~' \mid \E\';$	# F9 or ESC ^
pfk17 = '\E[21~' '\E&';	# F10 or ESC &
pfk18 = '\E[23~' '\E*';	# F11 or ESC *
pfk19 = '\E[24~' '\E(';	# F12 or ESC (

pfk21 = \B[26-' \BE-';	pfk20 = '\E[25~' '\E)';	# F13 or ESC)
pfk22 = '\E[31~' '\E='; #F17 or ESC = pfk23 = '\E[32~' '\E_'; #F18 or ESC_ pfk24 = '\E[33~' '\E+'; #F19 or ESC + pal = '\EOm' '\Eal'; #Numeric Minus or ESC a1 pa2 = '\EOl' '\Ea2'; #Numeric Comma or ESC a2 pa3 = '\E[34~' '\Ea3'; #F20 or ESC a3 escape = '\a']'; #Telnet escape help = '\a'e' '\E[28~'; #On-line help keymap = '\a'k'; #Put out the keymapping option = '\a'o'; #Set options lprt = '\approx p'; #Print screen (to a file) fm = '\approx p'; #Set Field Mark dup = '\approx d'; #Duplicate eh_v = '\Ev'; #Highlighting according to currently valid state eh_f = '\EF'; #Highlighting as for field eh_b = '\Eb'; #Blinking eh_r = '\Er'; #Reverse video eh_u = \Eu'; #Underscore ec_v = '\EV'; #Colour according to currently valid state ec_f = '\EF'; #Red ec_p = '\EB'; #Red ec_p = '\EB'; #Green ec_t = '\EF'; #Fink ec_g = '\EG'; #Green ec_t = '\ET'; #Yellow ec_n = '\ET'; #Yellow ec_n = '\EN'; #Neutral extended ASCII normal = '\an'; #Switch to normal ASCII mreset = '\a'g'; #Master reset		, , , , , , , , , , , , , , , , , , ,
pfk23 = '\E 32~' '\E_'; #F18 or ESC_ pfk24 = '\E 33~' '\E+'; #F19 or ESC + pal = '\EOM' '\Eal'; #Numeric Minus or ESC a1 pa2 = '\EOl' '\Ea2'; #Numeric Comma or ESC a2 pa3 = '\E 34~' '\Ea3'; #F20 or ESC a3 escape = '\B'; #On-line help keymap = '\A'; #Put out the keymapping option = '\A'o; #Print screen (to a file) fm = '\A'f; #Set Field Mark dup = '\A'f; #Duplicate eh_v = '\E'; #Highlighting according to currently valid state eh_f = \Ef'; #Highlighting as for field eh_b = \Eb'; #Reverse video eh_u = \Eu'; #Underscore ec_v = '\EV'; #Red ec_b = \EF; #Red ec_p = \EF; #Red ec_p = \EF; #Red ec_y = \EF'; #Fink ec_n = \EF'; #Fink extend = '\A'x'; #Fink extend = '\A'y';		
pfk24 = \E[33~' \E+';		
pa1 = \EOm' \Ea1'; # Numeric Minus or ESC a1 pa2 = \EOl' \Ea2'; # Numeric Comma or ESC a2 pa3 = \E[34-' \Ea3'; # F20 or ESC a3 escape = '^ ; # Telnet escape help = '^e' \E[28-'; # On-line help keymap = '^k'; # Put out the keymapping option = '^o'; # Set options lprt = '^p'; # Print screen (to a file) fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = \Ev'; # Highlighting according to currently valid state eh_f = \Ef'; # Highlighting as for field eh_b = \Eb'; # Blinking eh_r = \Er'; # Reverse video eh_u = \Eu'; # Underscore ec_v = \EV'; # Colour according to currently valid state ec_f = \EF; # Colour as for field ec_b = \EB'; # Blue ec_r = \ER'; # Red ec_p = \EP'; # Red ec_p = \EP'; # Pink ec_g = \EP'; # Green ec_t = \ET'; # Turquoise ec_y = \EY'; # Yellow ec_n = \EY'; # Yellow ec_n = \EY'; # Neutral extend = '^x'; # Switch to extended ASCII mormal = '^n'; # Master reset reshow = '^v';		
pa2 = \EOI' \Ea2'; # Numeric Comma or ESC a2 pa3 = \E[34~' \Ea3'; # F20 or ESC a3 escape = '^ ; # Telnet escape help = '^e' \E[28~'; # On-line help keymap = '^k; # Put out the keymapping option = '^o'; # Set options lprt = '^p'; # Print screen (to a file) fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = \Ev'; # Highlighting according to currently valid state eh_f = \Ef'; # Highlighting as for field eh_b = \Eb'; # Reverse video eh_u = \Eu'; # Underscore ec_v = \Ev'; # Colour according to currently valid state ec_f = \EF'; # Colour as for field ec_b = \EF'; # Red ec_p = \EF'; # Red ec_r = \EF'; # Red ec_r = \EF'; # Green ec_y = \EF'; # Green ec_y = \EF'; # Yellow ec_n = \ET'; # Yellow ec_n = \ET'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Master reset reshow = '^v';		
pa3 = \E[34~' \Ea3'; #F20 or ESC a3 escape = '^ ; #Telnet escape help = '^e' \E[28~'; #On-line help keymap = '^k'; #Put out the keymapping option = '^o'; #Set options lprt = '^p'; #Print screen (to a file) fm = '^f; #Set Field Mark dup = '^d'; #Duplicate eh_v = \Ev'; #Highlighting according to currently valid state eh_f = \Ef; #Highlighting as for field eh_b = \Eb'; #Reverse video eh_u = \Eu'; #Reverse video eh_u = \Eu'; #Underscore ec_v = \EV'; #Set Field Mark #Highlighting according to currently valid state eh_f = \Ef; #Reverse video eh_u = \Eu'; #Reverse video eh_u = \Eu'; #Reverse video ec_v = \EV'; #Reverse video ec_v = \EV'; #Reverse video ec_b = \EF; #Colour according to currently valid state ec_f = \EF; #Colour according to currently valid state ec_f = \EF; #Red ec_p = \EF; #Red ec_p = \EF'; #Red ec_p = \EF'; #Red ec_p = \EF'; #Fink ec_g = \EF'; #Fink ec_g = \EF'; #Fink ec_g = \EF'; #Fink ec_g = \EF'; #Fink ec_n = \EF'; #Fink extend = \Arx'; #Fink extend =	pa2 = '\EO1' '\Ea2';	# Numeric Comma or ESC a2
escape = '^]'; # Telnet escape help = '^e' ^E[28~'; # On-line help keymap = '^k'; # Put out the keymapping option = '^o'; # Set options lprt = '^p'; # Print screen (to a file) fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = ^E'; # Highlighting according to currently valid state eh_f = 'Ef'; # Highlighting as for field eh_b = ^E'; # Blinking eh_r = ^E'; # Reverse video eh_u = ^E'; # Underscore ec_v = ^EV'; # Colour according to currently valid state ec_f = ^EF; # Colour as for field ec_b = ^EF; # Red ec_p = ^EF; # Red ec_p = ^EF; # Red ec_r = ^EF; # Red ec_r = ^EF; # Fink ec_g = ^EF; # Fink ec_g = ^EF; # Furquoise ec_t = ^EF; # Yellow ec_n = ^EF'; # Yellow ec_n = ^EF'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset reshow = '^v';		# F20 or ESC a3
help = '^e' '\E[28~'; # On-line help keymap = '^k'; # Put out the keymapping option = '^o'; # Set options lprt = '^p'; # Print screen (to a file) fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = '\Ev'; # Highlighting according to currently valid state eh_f = '\Ef; # Highlighting as for field eh_b = '\Eb'; # Blinking eh_r = '\Er'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF; # Colour as for field ec_b = '\EF; # Red ec_r = '\ER'; # Red ec_r = '\ER'; # Red ec_p = '\EP'; # Pink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\ET'; # Yellow ec_n = '\ET'; # Yellow extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '\g'; # Master reset reshow = '\'v';	escape = '^]';	# Telnet escape
keymap = '^k'; # Put out the keymapping option = '^o'; # Set options lprt = '^p'; # Print screen (to a file) fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = '\Ev'; # Highlighting according to currently valid state eh_f = \Ef'; # Highlighting as for field eh_b = '\Eb'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EF'; # Red ec_p = '\EF'; # Red ec_p = '\EF'; # Red ec_r = '\EF'; # Fink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\ET'; # Yellow ec_n = '\ET'; # Yellow ec_n = '\ET'; # Neutral extend = '^x'; # Switch to extended ASCII mormal = '^n'; # Switch to normal ASCII mreset = '\g'; # Master reset reshow = '\v';		# On-line help
option = '^o'; # Set options lprt = '^p'; # Print screen (to a file) fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = '\Ev'; # Highlighting according to currently valid state eh_f = \Ef'; # Highlighting as for field eh_b = '\Eb'; # Blinking eh_r = \Er'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = \Ef'; # Colour as for field ec_b = '\EF'; # Red ec_r = '\EF'; # Red ec_r = '\EF'; # Red ec_r = '\EF'; # Pink ec_g = '\EG'; # Green ec_t = '\ET'; # Yellow ec_y = '\ET'; # Yellow ec_n = '\ET'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Master reset reshow = '^v';		# Put out the keymapping
fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = '\Ev'; # Highlighting according to currently valid state eh_f = '\Ef'; # Highlighting as for field eh_b = '\Eb'; # Blinking eh_r = '\Er'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Red ec_r = '\ER'; # Red ec_r = '\ER'; # Red ec_g = '\EG'; # Green ec_t = '\EF'; # Yellow ec_y = '\EY'; # Yellow ec_n = '\EN'; # Neutral mormal = '^n'; # Switch to normal ASCII mormal = '\n'; # Master reset	option = '^o';	
fm = '^f; # Set Field Mark dup = '^d'; # Duplicate eh_v = '\Ev'; # Highlighting according to currently valid state eh_f = '\Ef'; # Highlighting as for field eh_b = '\Eb'; # Blinking eh_r = '\Er'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Red ec_r = '\ER'; # Red ec_r = '\ER'; # Red ec_g = '\EG'; # Green ec_t = '\EF'; # Yellow ec_y = '\EY'; # Yellow ec_n = '\EN'; # Neutral mormal = '^n'; # Switch to normal ASCII mormal = '\n'; # Master reset		# Print screen (to a file)
eh_v = '\Ev'; # Highlighting according to currently valid state eh_f = '\Ef'; # Highlighting as for field eh_b = '\Eb'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Blue ec_r = '\ER'; # Red ec_p = '\ER'; # Red ec_g = '\EF'; # Fink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_t = '\ET'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '\g'; # Master reset reshow = '\v';	fm = '^f';	# Set Field Mark
eh_v = '\Ev', valid state eh_f = '\Ef'; # Highlighting as for field eh_b = '\Eb'; # Reverse video eh_v = '\Ev'; # Colour according to currently valid state ec_v = '\EV'; # Colour as for field ec_b = '\EF'; # Colour as for field ec_b = '\EB'; # Red ec_r = '\ER'; # Red ec_r = '\ER'; # Pink ec_g = '\EC'; # Green ec_t = '\ET'; # Turquoise ec_t = '\ET'; # Yellow ec_n = '\EY'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '\n'; # Switch to normal ASCII mreset = '\c'g'; # Master reset reshow = '\c'v';	dup = '^d';	# Duplicate
eh_b = '\Eb'; # Blinking eh_r = '\Er'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Blue ec_r = '\ER'; # Red ec_p = '\ER'; # Red ec_p = '\EP'; # Pink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\ET'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset	eh_v = '\Ev';	
eh_b = '\Eb'; # Blinking eh_r = '\Er'; # Reverse video eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Blue ec_r = '\ER'; # Red ec_p = '\ER'; # Red ec_p = '\EP'; # Pink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\ET'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset	eh_f = '\Ef';	# Highlighting as for field
eh_u = '\Eu'; # Underscore ec_v = '\EV'; # Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Blue ec_r = '\ER'; # Red ec_p = '\EP'; # Pink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\EY'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset		
# Colour according to currently valid state ec_f = '\EF'; # Colour as for field ec_b = '\EB'; # Blue ec_r = '\ER'; # Red ec_p = '\EP'; # Pink ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\EY'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = ''g'; # Master reset	eh_r = '\Er';	# Reverse video
state cc_f = '\EF';	eh_u = '\Eu';	# Underscore
ec_b = '\EB';	ec_v = '\EV';	·
ec_r = '\ER';	ec_f = '\EF';	# Colour as for field
ec_p = '\EP';	$ec_b = \EB';$	# Blue
ec_g = '\EG'; # Green ec_t = '\ET'; # Turquoise ec_y = '\EY'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset reshow = '^v';	$ec_r = \ER';$	# Red
ec_t = '\ET';	ec_p = '\EP';	# Pink
ec_y = '\EY'; # Yellow ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset reshow = '^v';	ec_g = '\EG';	# Green
ec_n = '\EN'; # Neutral extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset reshow = '^v';	ec_t = '\ET';	# Turquoise
extend = '^x'; # Switch to extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset reshow = '^v';	ec_y = '\EY';	# Yellow
extended ASCII normal = '^n'; # Switch to normal ASCII mreset = '^g'; # Master reset reshow = '^v';	$ec_n = \EN';$	# Neutral
mreset = '^g'; # Master reset reshow = '^v';	[] ·	
reshow = '^v';	normal = '^n';	# Switch to normal ASCII
reshow = '^v';	mreset = '^g';	# Master reset
reset = '^t';		
	reset = '^t';	

ESC Sequences

l -	
Key Sequence	Meaning
ESC(B	Set to US ASCII character set
ESC(A	UK National
ESC(0	Graphic
ESC[M	Move the cursor up in scrolling region.
ESC[nA	Move the cursor up.
ESCD'LF'	Move the cursor down in scrolling region.
ESC[nB	Move the cursor down
ESC[nC	Move the cursor right
ESC[nD	Move the cursor left
ESCE	Move cursor to start of next line
ESC[r;cH	Move the cursor to row r, column c.
ESC7	Save the cursor column position and character attributes
ESC8	Restore the cursor column position and character attributes
ESC[0K	Erase from the cursor to the end of the line
ESC[1K	erase from the beginning of the line to the cursor.
ESC[2K	Erase the line
ESC[0J	Erase from the cursor to the end of the screen.
ESC[1J	erase from the bottom of the screen to the cursor
ESC[2J	Erase the screen
ESC[t;br	Defines a scrolling region where the top line = t and the bottom line = b.
ESC[?6h	Set cursor position 0,0 equal to the upper left corner of the scrolling region.
ESC[?61	Set cursor position 0,0 equal to upper left corner of the defined scrolling region.
ESC[0m	Normal characters
ESC[1m	Bold characters
ESC[4m	Underline characters.
ESC[5m	Blinking characters
ESC[7m	reverse video characters
ESC#3	Double height characters, top half.
ESC#4	Double height characters, bottom
ESC#5	Single width characters
ESC#6	Double width characters
ESC[?31	Set terminal to 132 column mode

ESC[?3h	Set terminal to 80 column mode
ESCH	Set tab stop at current column position
ESC[g	Clear tab stop at current column.
ESC[3g	Clear all tab stops.