Xterm Control Sequences

Edward Moy

University of California, Berkeley

Revised by

Stephen Gildea

X Consortium (1994)

Thomas Dickey

XFree86 Project (1996-1999)

Definitions

$\boxed{c}$ The literal character *c*.

C A single (required) character.

P_s A single (usually optional) numeric parameter, composed of one of more digits.

P_m A multiple numeric parameter composed of any number of single numeric parameters, separated by $\boxed{;}$ character(s). Individual values for the parameters are listed with P_s .

P_t A text parameter composed of printable characters.

C1 (8-Bit) Control Characters

The *xterm* program recognizes both 8-bit and 7-bit control characters. It generates 7-bit controls (by default) or 8-bit if S8C1T is enabled. The following pairs of 7-bit and 8-bit control characters are equivalent:

ESC D

Index (IND: 0x84)

ESC E

Next Line (NEL: 0x85)

ESC H

Tab Set (HTS: 0x88)

ESC M

Reverse Index (RI: 0x8d)

ESC N

Single Shift Select of G2 Character Set (SS2: 0x8e): affects next character only

ESC O

Single Shift Select of G3 Character Set (SS3: 0x8f): affects next character only

ESC P

Device Control String (DCS: 0x90)

ESC V

Start of Guarded Area (SPA: 0x96)

ESC W
: End of Guarded Area (EPA: 0x97)

ESC X
: Start of String (SOS: 0x98)

ESC Z
: Return Terminal ID (DECID: 0x9a). Obsolete form of CSI c (DA).

ESC [
: Control Sequence Introducer (CSI: 0x9b)

ESC \
: String Terminator (ST: 0x9c)

ESC]
: Operating System Command (OSC: 0x9d)

ESC ^
: Privacy Message (PM: 0x9e)

ESC _
: Application Program Command (APC: 0x9f)

These control characters are used in the vtXXX emulation.

VT100 Mode

Most of these control sequences are standard VT102 control sequences, but there is support for later DEC VT terminals (i.e., VT220 and VT320), as well as ISO 6429 and *aixterm* color controls. VT102 features not supported are auto-repeat and blinking characters. There are additional control sequences to provide *xterm*-dependent functions, such as the scrollbar or window size. Where the function is specified by DEC or ISO 6429, the code assigned to it is given in parentheses. The escape codes to designate and invoke character sets are specified by ISO 2022; see that document for a discussion of character sets.

Single-character functions

BEL	Bell (Ctrl-G)
BS	Backspace (Ctrl-H)
CR	Carriage Return (Ctrl-M)
ENQ	Return Terminal Status (Ctrl-E). Default response is the terminal name, e.g., "xterm", but may be overridden by a resource *answerbackString*.
FF	Form Feed or New Page (NP) (Ctrl-L) same as LF
LF	Line Feed or New Line (NL) (Ctrl-J)
SO	Shift Out (Ctrl-N) → Switch to Alternate Character Set: invokes the G1 character set.
SP	Space.
TAB	Horizontal Tab (HT) (Ctrl-I)
VT	Vertical Tab (Ctrl-K) same as LF
SI	Shift In (Ctrl-O) → Switch to Standard Character Set: invokes the G0 character set (the default).

Controls beginning with ESC (other than those where ESC is part of a 7-bit equivalent to 8-bit C1 controls), ordered by the final character(s).

ESC SP F	7-bit controls (S7C1T).
ESC SP G	8-bit controls (S8C1T).
ESC SP L	Set ANSI conformance level 1 (vt100, 7-bit controls).
ESC SP M	Set ANSI conformance level 2 (vt200).
ESC SP N	Set ANSI conformance level 3 (vt300).
ESC # 3	DEC double-height line, top half (DECDHL)
ESC # 4	DEC double-height line, bottom half (DECDHL)
ESC # 5	DEC single-width line (DECSWL)
ESC # 6	DEC double-width line (DECDWL)
ESC # 8	DEC Screen Alignment Test (DECALN)
ESC % @	Select default character set, ISO 8859-1 (ISO 2022)
ESC % G	Select UTF-8 character set (ISO 2022)
ESC (*C*	Designate G0 Character Set (ISO 2022)
ESC) *C*	Designate G1 Character Set (ISO 2022)
ESC * *C*	Designate G2 Character Set (ISO 2022)
ESC + *C*	Designate G3 Character Set (ISO 2022)

Final character *C* for designating character sets (0, A and B apply to VT100 and up, the remainder to VT220 and up):

- *C* = 0 → DEC Special Character and Line Drawing Set
- *C* = A → United Kingdom (UK)
- *C* = B → United States (USASCII)
- *C* = 4 → Dutch
- *C* = C or 5 → Finnish
- *C* = R → French
- *C* = Q → French Canadian
- *C* = K → German
- *C* = Y → Italian
- *C* = E or 6 → Norwegian/Danish
- *C* = Z → Spanish
- *C* = H or 7 → Swedish
- *C* = = → Swiss

ESC 7	Save Cursor (DECSC)
ESC 8	Restore Cursor (DECRC)
ESC =	Application Keypad (DECPAM)
ESC >	Normal Keypad (DECPNM)
ESC F	Cursor to lower left corner of screen (if enabled by the **hpLowerleftBugCompat** resource).
ESC c	Full Reset (RIS)

ESC l	Memory Lock (per HP terminals). Locks memory above the cursor.
ESC m	Memory Unlock (per HP terminals)
ESC n	Invoke the G2 Character Set (LS2)
ESC o	Invoke the G3 Character Set (LS3)
ESC |	Invoke the G3 Character Set as GR (LS3R). Has no visible effect in *xterm*.
ESC }	Invoke the G2 Character Set as GR (LS2R). Has no visible effect in *xterm*.
ESC ~	Invoke the G1 Character Set as GR (LS1R). Has no visible effect in *xterm*.

Application Program-Control functions

APC P_t ST — *xterm* implements no APC functions; P_t is ignored. P_t need not be printable characters.

Device-Control functions

DCS P_s ; P_s | P_t ST — User-Defined Keys (DECUDK). The first parameter:

- P_s = 0 → Clear all UDK definitions before starting (default)
- P_s = 1 → Erase Below (default)

The second parameter:

- P_s = 0 → Lock the keys (default)
- P_s = 1 → Do not lock.

The third parameter is a ';'-separated list of strings denoting the key-code separated by a '/' from the hex-encoded key value. The key codes correspond to the DEC function-key codes (e.g., F6=17).

DCS $ q P_t ST — Request Status String (DECRQSS). The string following the "q" is one of the following:

- " q → DECSCA
- " p → DECSCL
- r → DECSTBM
- m → SGR

xterm responds with DCS 1 $ r P_t ST for valid requests, replacing the P_t with the corresponding CSI string, or DCS 0 $ r P_t ST for invalid requests.

Functions using CSI, ordered by the final character(s)

CSI P_s @	Insert P_s (Blank) Character(s) (default = 1) (ICH)
CSI P_s A	Cursor Up P_s Times (default = 1) (CUU)
CSI P_s B	Cursor Down P_s Times (default = 1) (CUD)
CSI P_s C	Cursor Forward P_s Times (default = 1) (CUF)
CSI P_s D	Cursor Backward P_s Times (default = 1) (CUB)
CSI P_s E	Cursor Next Line P_s Times (default = 1) (CNL)
CSI P_s F	Cursor Preceding Line P_s Times (default = 1) (CPL)
CSI P_s G	Cursor Character Absolute [column] (default = [row,1]) (CHA)
CSI P_s ; P_s H	Cursor Position [row;column] (default = [1,1]) (CUP)
CSI P_s I	Cursor Forward Tabulation P_s tab stops (default = 1) (CHT)

CSI P_s J	Erase in Display (ED) P_s = 0 → Erase Below (default) P_s = 1 → Erase Above P_s = 2 → Erase All P_s = 3 → Erase Saved Lines (xterm)
CSI ? P_s J	Erase in Display (DECSED) P_s = 0 → Selective Erase Below (default) P_s = 1 → Selective Erase Above P_s = 2 → Selective Erase All
CSI P_s K	Erase in Line (EL) P_s = 0 → Erase to Right (default) P_s = 1 → Erase to Left P_s = 2 → Erase All
CSI ? P_s K	Erase in Line (DECSEL) P_s = 0 → Selective Erase to Right (default) P_s = 1 → Selective Erase to Left P_s = 2 → Selective Erase All
CSI P_s L	Insert P_s Line(s) (default = 1) (IL)
CSI P_s M	Delete P_s Line(s) (default = 1) (DL)
CSI P_s P	Delete P_s Character(s) (default = 1) (DCH)
CSI P_s S	Scroll up P_s lines (default = 1) (SU)
CSI P_s T	Scroll down P_s lines (default = 1) (SD)
CSI P_s ; P_s ; P_s ; P_s ; P_s T	Initiate hilite mouse tracking. Parameters are [func;startx;starty;firstrow;lastrow]. See the section **Mouse Tracking**.
CSI P_s X	Erase P_s Character(s) (default = 1) (ECH)
CSI P_s Z	Cursor Backward Tabulation P_s tab stops (default = 1) (CBT)
CSI P_m `	Character Position Absolute [column] (default = [row,1]) (HPA)
CSI P_s b	Repeat the preceding graphic character P_s times (REP)
CSI P_s c	Send Device Attributes (Primary DA) P_s = 0 , 1 or omitted → request attributes from terminal. The response depends on the **decTerminalId** resource setting. → CSI ? 1 ; 2 c (“VT100 with Advanced Video Option”) → CSI ? 1 ; 0 c (“VT101 with No Options”) → CSI ? 6 c (“VT102”) → CSI ? 6 0 ; 1 ; 2 ; 6 ; 8 ; 9 ; 1 5 ; c (“VT220”) The VT100-style response parameters do not mean anything by themselves. VT220 parameters do, telling the host what features the terminal supports: → 1 132-columns → 2 Printer

→ $\boxed{6}$ Selective erase
→ $\boxed{8}$ User-defined keys
→ $\boxed{9}$ National replacement character sets
→ $\boxed{1}\boxed{5}$ Technical characters

$\boxed{\text{CSI}}\boxed{>}P_s\boxed{c}$ Send Device Attributes (Secondary DA)
$P_s = \boxed{0}$, 1 or omitted → request the terminal's identification code. The response depends on the **decTerminalId** resource setting. It should apply only to VT220 and up, but *xterm* extends this to VT100.
→ $\boxed{\text{CSI}}P_p\boxed{;}P_v\boxed{;}P_c\boxed{c}$
where P_p denotes the terminal type
→ $\boxed{0}$ ("VT100")
→ $\boxed{1}$ ("VT220")
and P_v is the firmware version (for *xterm*, this is the XFree86 patch number, starting with 95). In a DEC terminal, P_c indicates the ROM cartridge registration number and is always zero.

$\boxed{\text{CSI}}P_m\boxed{d}$ Line Position Absolute [row] (default = [1,column]) (VPA)

$\boxed{\text{CSI}}P_s\boxed{;}P_s\boxed{f}$ Horizontal and Vertical Position [row;column] (default = [1,1]) (HVP)

$\boxed{\text{CSI}}P_s\boxed{g}$ Tab Clear (TBC)
$P_s = \boxed{0}$ → Clear Current Column (default)
$P_s = \boxed{3}$ → Clear All

$\boxed{\text{CSI}}P_m\boxed{h}$ Set Mode (SM)
$P_s = \boxed{2}$ → Keyboard Action Mode (AM)
$P_s = \boxed{4}$ → Insert Mode (IRM)
$P_s = \boxed{1}\boxed{2}$ → Send/receive (SRM)
$P_s = \boxed{2}\boxed{0}$ → Automatic Newline (LNM)

$\boxed{\text{CSI}}\boxed{?}P_m\boxed{h}$ DEC Private Mode Set (DECSET)
$P_s = \boxed{1}$ → Application Cursor Keys (DECCKM)
$P_s = \boxed{2}$ → Designate USASCII for character sets G0-G3 (DECANM), and set VT100 mode.
$P_s = \boxed{3}$ → 132 Column Mode (DECCOLM)
$P_s = \boxed{4}$ → Smooth (Slow) Scroll (DECSCLM)
$P_s = \boxed{5}$ → Reverse Video (DECSCNM)
$P_s = \boxed{6}$ → Origin Mode (DECOM)
$P_s = \boxed{7}$ → Wraparound Mode (DECAWM)
$P_s = \boxed{8}$ → Auto-repeat Keys (DECARM)
$P_s = \boxed{9}$ → Send Mouse X & Y on button press. See the section **Mouse Tracking**.
$P_s = \boxed{1}\boxed{8}$ → Print form feed (DECPFF)
$P_s = \boxed{1}\boxed{9}$ → Set print extent to full screen (DECPEX)
$P_s = \boxed{2}\boxed{5}$ → Show Cursor (DECTCEM)
$P_s = \boxed{3}\boxed{0}$ → Show scrollbar (rxvt).
$P_s = \boxed{3}\boxed{5}$ → Enable shifted key-functions (rxvt).

P_s = 3 8 → Enter Tektronix Mode (DECTEK)
P_s = 4 0 → Allow 80 ↔ 132 Mode
P_s = 4 1 → *more*(1) fix (see *curses* resource)
P_s = 4 2 → Enable Nation Replacement Character sets (DECNRCM)
P_s = 4 4 → Turn On Margin Bell
P_s = 4 5 → Reverse-wraparound Mode
P_s = 4 6 → Start Logging (normally disabled by a compile-time option)
P_s = 4 7 → Use Alternate Screen Buffer (unless disabled by the **titeInhibit** resource)
P_s = 6 6 → Application keypad (DECNKM)
P_s = 6 7 → Backarrow key sends delete (DECBKM)
P_s = 1 0 0 0 → Send Mouse X & Y on button press and release. See the section **Mouse Tracking**.
P_s = 1 0 0 1 → Use Hilite Mouse Tracking.
P_s = 1 0 0 2 → Use Cell Motion Mouse Tracking.
P_s = 1 0 0 3 → Use All Motion Mouse Tracking.
P_s = 1 0 1 0 → Scroll to bottom on tty output (rxvt).
P_s = 1 0 1 1 → Scroll to bottom on key press (rxvt).
P_s = 1 0 3 5 → Enable special modifiers for Alt and NumLock keys.
P_s = 1 0 3 6 → Send ESC when Meta modifies a key (enables the **metaSendsEscape** resource).
P_s = 1 0 3 7 → Send DEL from the editing-keypad Delete key
P_s = 1 0 4 7 → Use Alternate Screen Buffer (unless disabled by the **titeInhibit** resource)
P_s = 1 0 4 8 → Save cursor as in DECSC (unless disabled by the **titeInhibit** resource)
P_s = 1 0 4 9 → Save cursor as in DECSC and use Alternate Screen Buffer, clearing it first (unless disabled by the **titeInhibit** resource)
P_s = 1 0 5 1 → Set Sun function-key mode.
P_s = 1 0 5 2 → Set HP function-key mode.
P_s = 1 0 6 0 → Set legacy keyboard emulation (X11R6).
P_s = 1 0 6 1 → Set Sun/PC keyboard emulation of VT220 keyboard.

CSI P_m i — Media Copy (MC)
P_s = 0 → Print screen (default)
P_s = 4 → Turn off printer controller mode
P_s = 5 → Turn on printer controller mode

CSI ? P_m i — Media Copy (MC, DEC-specific)
P_s = 1 → Print line containing cursor
P_s = 4 → Turn off autoprint mode
P_s = 5 → Turn on autoprint mode
P_s = 1 0 → Print composed display, ignores DECPEX

P_s = 1 1 → Print all pages

CSI P_m l — Reset Mode (RM)

P_s = 2 → Keyboard Action Mode (AM)
P_s = 4 → Replace Mode (IRM)
P_s = 1 2 → Send/receive (SRM)
P_s = 2 0 → Normal Linefeed (LNM)

CSI ? P_m l — DEC Private Mode Reset (DECRST)

P_s = 1 → Normal Cursor Keys (DECCKM)
P_s = 2 → Designate VT52 mode (DECANM).
P_s = 3 → 80 Column Mode (DECCOLM)
P_s = 4 → Jump (Fast) Scroll (DECSCLM)
P_s = 5 → Normal Video (DECSCNM)
P_s = 6 → Normal Cursor Mode (DECOM)
P_s = 7 → No Wraparound Mode (DECAWM)
P_s = 8 → No Auto-repeat Keys (DECARM)
P_s = 9 → Don't Send Mouse X & Y on button press
P_s = 1 8 → Don't print form feed (DECPFF)
P_s = 1 9 → Limit print to scrolling region (DECPEX)
P_s = 2 5 → Hide Cursor (DECTCEM)
P_s = 3 0 → Don't show scrollbar (rxvt).
P_s = 3 5 → Disable shifted key-functions (rxvt).
P_s = 4 0 → Disallow 80 ↔ 132 Mode
P_s = 4 1 → No *more*(1) fix (see *curses* resource)
P_s = 4 2 → Disable Nation Replacement Character sets (DECNRCM)
P_s = 4 4 → Turn Off Margin Bell
P_s = 4 5 → No Reverse-wraparound Mode
P_s = 4 6 → Stop Logging (normally disabled by a compile-time option)
P_s = 4 7 → Use Normal Screen Buffer
P_s = 6 6 → Numeric keypad (DECNKM)
P_s = 6 7 → Backarrow key sends backspace (DECBKM)
P_s = 1 0 0 0 → Don't Send Mouse X & Y on button press and release. See the section **Mouse Tracking**.
P_s = 1 0 0 1 → Don't Use Hilite Mouse Tracking
P_s = 1 0 0 2 → Don't Use Cell Motion Mouse Tracking
P_s = 1 0 0 3 → Don't Use All Motion Mouse Tracking
P_s = 1 0 1 0 → Don't scroll to bottom on tty output (rxvt).
P_s = 1 0 1 1 → Don't scroll to bottom on key press (rxvt).
P_s = 1 0 3 5 → Disable special modifiers for Alt and NumLock keys.
P_s = 1 0 3 6 → Don't send ESC when Meta modifies a key (disables the **metaSendsEscape** resource).
P_s = 1 0 3 7 → Send VT220 Remove from the editing-keypad Delete key

P_s = 1 0 4 7 → Use Normal Screen Buffer, clearing screen first if in the Alternate Screen (unless disabled by the **titeInhibit** resource)
P_s = 1 0 4 8 → Restore cursor as in DECRC (unless disabled by the **titeInhibit** resource)
P_s = 1 0 4 9 → Use Normal Screen Buffer and restore cursor as in DECRC (unless disabled by the **titeInhibit** resource)
P_s = 1 0 5 1 → Reset Sun function-key mode.
P_s = 1 0 5 2 → Reset HP function-key mode.
P_s = 1 0 6 0 → Reset legacy keyboard emulation (X11R6).
P_s = 1 0 6 1 → Reset Sun/PC keyboard emulation of VT220 keyboard.

CSI P_m m — Character Attributes (SGR)
P_s = 0 → Normal (default)
P_s = 1 → Bold
P_s = 4 → Underlined
P_s = 5 → Blink (appears as Bold)
P_s = 7 → Inverse
P_s = 8 → Invisible (hidden)
P_s = 2 2 → Normal (neither bold nor faint)
P_s = 2 4 → Not underlined
P_s = 2 5 → Steady (not blinking)
P_s = 2 7 → Positive (not inverse)
P_s = 2 8 → Visible (not hidden)
P_s = 3 0 → Set foreground color to Black
P_s = 3 1 → Set foreground color to Red
P_s = 3 2 → Set foreground color to Green
P_s = 3 3 → Set foreground color to Yellow
P_s = 3 4 → Set foreground color to Blue
P_s = 3 5 → Set foreground color to Magenta
P_s = 3 6 → Set foreground color to Cyan
P_s = 3 7 → Set foreground color to White
P_s = 3 9 → Set foreground color to default (original)
P_s = 4 0 → Set background color to Black
P_s = 4 1 → Set background color to Red
P_s = 4 2 → Set background color to Green
P_s = 4 3 → Set background color to Yellow
P_s = 4 4 → Set background color to Blue
P_s = 4 5 → Set background color to Magenta
P_s = 4 6 → Set background color to Cyan
P_s = 4 7 → Set background color to White
P_s = 4 9 → Set background color to default (original).

If 16-color support is compiled, the following apply. Assume that *xterm*'s resources are set so that the ISO color codes are the first 8 of a set of 16. Then the *aixterm* colors are the bright versions of the ISO colors:

P_s = 9 0 → Set foreground color to Black
P_s = 9 1 → Set foreground color to Red
P_s = 9 2 → Set foreground color to Green
P_s = 9 3 → Set foreground color to Yellow
P_s = 9 4 → Set foreground color to Blue
P_s = 9 5 → Set foreground color to Magenta
P_s = 9 6 → Set foreground color to Cyan
P_s = 9 7 → Set foreground color to White
P_s = 1 0 0 → Set background color to Black
P_s = 1 0 1 → Set background color to Red
P_s = 1 0 2 → Set background color to Green
P_s = 1 0 3 → Set background color to Yellow
P_s = 1 0 4 → Set background color to Blue
P_s = 1 0 5 → Set background color to Magenta
P_s = 1 0 6 → Set background color to Cyan
P_s = 1 0 7 → Set background color to White

If *xterm* is compiled with the 16-color support disabled, it supports the following, from *rxvt*:

P_s = 1 0 0 → Set foreground and background color to default

If 88- or 256-color support is compiled, the following apply.

P_s = 3 8 ; 5 ; P_s → Set background color to P_s
P_s = 4 8 ; 5 ; P_s → Set foreground color to P_s

CSI P_s n — Device Status Report (DSR)

P_s = 5 → Status Report CSI 0 n ("OK")
P_s = 6 → Report Cursor Position (CPR) [row;column] as CSI *r* ; *c* R

CSI ? P_s n — Device Status Report (DSR, DEC-specific)

P_s = 6 → Report Cursor Position (CPR) [row;column] as CSI ? *r* ; *c* R (assumes page is zero).
P_s = 1 5 → Report Printer status as CSI ? 1 0 n (ready) or CSI ? 1 1 n (not ready)
P_s = 2 5 → Report UDK status as CSI ? 2 0 n (unlocked) or CSI ? 2 1 n (locked)
P_s = 2 6 → Report Keyboard status as CSI ? 2 7 ; 1 ; 0 ; 0 n (North American)

CSI ? 5 3 n Locator available, if compiled-in, or
CSI ? 5 0 n No Locator, if not.

CSI ! p — Soft terminal reset (DECSTR)

CSI P_s ; P_s " p — Set conformance level (DECSCL) Valid values for the first parameter:
P_s = 6 1 → VT100
P_s = 6 2 → VT200
P_s = 6 3 → VT300 Valid values for the second parameter:
P_s = 0 → 8-bit controls
P_s = 1 → 7-bit controls (always set for VT100)
P_s = 2 → 8-bit controls

CSI P_s " q — Select character protection attribute (DECSCA). Valid values for the parameter:
P_s = 0 → DECSED and DECSEL can erase (default)
P_s = 1 → DECSED and DECSEL cannot erase
P_s = 2 → DECSED and DECSEL can erase

CSI P_s ; P_s r — Set Scrolling Region [top;bottom] (default = full size of window) (DECSTBM)

CSI ? P_m r — Restore DEC Private Mode Values. The value of P_s previously saved is restored. P_s values are the same as for DECSET.

CSI ? P_m s — Save DEC Private Mode Values. P_s values are the same as for DECSET.

CSI P_s ; P_s ; P_s t — Window manipulation (from *dtterm*, as well as extensions). Valid values for the first (and any additional parameters) are:
P_s = 1 → De-iconify window.
P_s = 2 → Iconify window.
P_s = 3 ; *x ; y* → Move window to [x, y].
P_s = 4 ; *height ; width* → Resize the *xterm* window to height and width in pixels.
P_s = 5 → Raise the *xterm* window to the front of the stacking order.
P_s = 6 → Lower the *xterm* window to the bottom of the stacking order.
P_s = 7 → Refresh the *xterm* window.
P_s = 8 ; *height ; width* → Resize the text area to [height;width] in characters.
P_s = 9 ; 0 → Restore maximized window.
P_s = 9 ; 1 → Maximize window (i.e., resize to screen size).
P_s = 1 1 → Report *xterm* window state. If the *xterm* window is open (non-iconified), it returns CSI 1 t . If the *xterm* window is iconified, it returns CSI 2 t .
P_s = 1 3 → Report *xterm* window position as CSI 3 ; *x* ; *y* t
P_s = 1 4 → Report *xterm* window in pixels as CSI 4 ; *height* ; *width* t
P_s = 1 8 → Report the size of the text area in characters as CSI 8 ; *height* ; *width* t
P_s = 1 9 → Report the size of the screen in characters as CSI 9 ; *height* ; *width* t
P_s = 2 0 → Report *xterm* window's icon label as OSC L *label* ST
P_s = 2 1 → Report *xterm* window's title as OSC 1 *title* ST
P_s >= 2 4 → Resize to P_s lines (DECSLPP)

CSI P_t ; P_l ; P_b ; P_r ` w
Enable Filter Rectangle (DECEFR)
Parameters are [top;left;bottom;right].
Defines the coordinates of a filter rectangle and activates it. Anytime the locator is detected outside of the filter rectangle, an outside rectangle event is generated and the rectangle is disabled. Filter rectangles are always treated as "one-shot" events. Any parameters that are omitted default to the current locator position. If all parameters are omitted, any locator motion will be reported. DECELR always cancels any prevous rectangle definition.

CSI P_s x
Request Terminal Parameters (DECREQTPARM)

CSI P_s ; P_u ` z
Enable Locator Reporting (DECELR)
Valid values for the first parameter:
P_s = 0 → Locator disabled (default)
P_s = 1 → Locator enabled
P_s = 2 → Locator enabled for one report, then disabled
The second parameter specifies the coordinate unit for locator reports.
Valid values for the second parameter:
P_u = 0 or omitted → default to character cells
P_u = 1 → device physical pixels
P_u = 2 → character cells

CSI P_m ` {
Select Locator Events (DECSLE)
Valid values for the first (and any additional parameters) are:
P_s = 0 → only respond to explicit host requests (DECRQLP)
(default) also cancels any filter rectangle
P_s = 1 → report button down transitions
P_s = 2 → do not report button down transitions
P_s = 3 → report button up transitions
P_s = 4 → do not report button up transitions

CSI P_s ` |
Request Locator Position (DECRQLP)
Valid values for the parameter are:
P_s = 0 , 1 or omitted → transmit a single DECLRP locator report

If Locator Reporting has been enabled by a DECELR, xterm will respond with a DECLRP Locator Report. This report is also generated on button up and down events if they have been enabled with a DECSLE, or when the locator is detected outside of a filter rectangle, if filter rectangles have been enabled with a DECEFR.

→ CSI P_e ; P_b ; P_r ; P_c ; P_p & w

Parameters are [event;button;row;column;page].
Valid values for the event:

P_e = 0 → locator unavailable - no other parameters sent
P_e = 1 → request - xterm received a DECRQLP
P_e = 2 → left button down
P_e = 3 → left button up
P_e = 4 → middle button down
P_e = 5 → middle button up
P_e = 6 → right button down
P_e = 7 → right button up
P_e = 8 → M4 button down
P_e = 9 → M4 button up
P_e = 1 0 → locator outside filter rectangle
‘‘button’’ parameter is a bitmask indicating which buttons are pressed:
P_b = 0 → no buttons down
P_b & 1 → right button down
P_b & 2 → middle button down
P_b & 4 → left button down
P_b & 8 → M4 button down
‘‘row’’ and ‘‘column’’ parameters are the coordinates of the locator position in the xterm window, encoded as ASCII decimal.
The ‘‘page’’ parameter is not used by xterm, and will be omitted.

Operating System Controls

OSC P_s ; P_t ST
OSC P_s ; P_t BEL

Set Text Parameters. For colors and font, if P_t is a "?", the control sequence elicits a response which consists of the control sequence which would set the corresponding value. The *dtterm* control sequences allow you to determine the icon name and window title.
P_s = 0 → Change Icon Name and Window Title to P_t
P_s = 1 → Change Icon Name to P_t
P_s = 2 → Change Window Title to P_t
P_s = 3 → Set X property on top-level window. P_t should be in the form "*prop=value*", or just "*prop*" to delete the property
P_s = 4 ; *c ; name* → Change Color #*c* to *cname*. Any number of *c name* pairs may be given.
P_s = 1 0 → Change color names starting with text foreground to P_t (a list of one or more color names or RGB specifications, separated by semicolon, up to eight, as per *XParseColor*).
P_s = 1 1 → Change colors starting with text background to P_t
P_s = 1 2 → Change colors starting with text cursor to P_t
P_s = 1 3 → Change colors starting with mouse foreground to P_t
P_s = 1 4 → Change colors starting with mouse background to P_t

P_s = 1 5 → Change colors starting with Tek foreground to P_t
P_s = 1 6 → Change colors starting with Tek background to P_t
P_s = 1 7 → Change colors starting with highlight to P_t
P_s = 4 6 → Change Log File to P_t (normally disabled by a compile-time option)
P_s = 5 0 → Set Font to P_t If P_t begins with a "#", index in the font menu, relative (if the next character is a plus or minus sign) or absolute. A number is expected but not required after the sign (the default is the current entry for relative, zero for absolute indexing).

Privacy Message

PM P_t ST — *xterm* implements no PM functions; P_t is ignored. P_t need not be printable characters.

PC-Style Function Keys

If *xterm* does minimal translation of the function keys, it usually does this with a PC-style keyboard, so PC-style function keys result. Sun keyboards are similar to PC keyboards. Both have cursor and scrolling operations printed on the keypad, which duplicate the smaller cursor and scrolling keypads.

X does not predefine NumLock (used for VT220 keyboards) or Alt (used as an extension for the Sun/PC keyboards) as modifiers. These keys are recognized as modifiers when enabled by the **numLock** resource, or by the DECSET 1 0 3 5 control sequence.

The cursor keys transmit the following escape sequences depending on the mode specified via the **DECCKM** escape sequence.

Key	Normal	Application
Cursor Up	CSI A	SS3 A
Cursor Down	CSI B	SS3 B
Cursor Right	CSI C	SS3 C
Cursor Left	CSI D	SS3 D

The application keypad transmits the following escape sequences depending on the mode specified via the **DECPNM** and **DECPAM** escape sequences. Use the NumLock key to override the application mode.

Not all keys are present on the Sun/PC keypad (e.g., PF1, Tab), but are supported by the program.

Key	Numeric	Application	Terminfo	Termcap
Space	SP	SS3 SP	-	-
Tab	TAB	SS3 I	-	-
Enter	CR	SS3 M	kent	@8
PF1	SS3 P	SS3 P	kf1	k1
PF2	SS3 Q	SS3 Q	kf2	k2
PF3	SS3 R	SS3 R	kf3	k3
PF4	SS3 S	SS3 S	kf4	k4
* (multiply)	*	SS3 j	-	-
+ (add)	+	SS3 k	-	-
, (comma)	,	SS3 l	-	-
- (minus)	-	SS3 m	-	-
. (Delete)	.	SS3 3 ~	-	-
/ (divide)	/	SS3 o	-	-
0 (Insert)	0	CSI 2 ~	-	-
1 (End)	1	CSI 4 ~	kc1	K4
2 (DownArrow)	2	CSI B	-	-

3 (PageDown)	**3**	CSI **5** ~	**kc3**	**K5**
4 (LeftArrow)	**4**	CSI **D**	**-**	**-**
5 (Begin)	**5**	CSI **E**	**kb2**	**K2**
6 (RightArrow)	**6**	CSI **C**	**-**	**-**
7 (Home)	**7**	CSI **1** ~	**ka1**	**K1**
8 (UpArrow)	**8**	CSI **A**	**-**	**-**
9 (PageUp)	**9**	CSI **5** ~	**ka3**	**K3**
= (equal)	**=**	SS3 **X**	**-**	**-**

They also provide 12 function keys, as well as a few other special-purpose keys.

Key	Escape Sequence
F1	SS3 **P**
F2	SS3 **Q**
F3	SS3 **R**
F4	SS3 **S**
F5	CSI **1 5** ~
F6	CSI **1 7** ~
F7	CSI **1 8** ~
F8	CSI **1 9** ~
F9	CSI **2 0** ~
F10	CSI **2 1** ~
F11	CSI **2 3** ~
F12	CSI **2 4** ~

Older versions of *xterm* implement different escape sequences for F1 through F4. These can be activated by setting the *oldXtermFKeys* resource. However, since they do not correspond to any hardware terminal, they have been deprecated. (The DEC VT220 reserves F1 through F5 for local functions such as **Setup**).

Key	Escape Sequence
F1	CSI **1 1** ~
F2	CSI **1 2** ~
F3	CSI **1 3** ~
F4	CSI **1 4** ~

In normal mode, i.e., a Sun/PC keyboard when the **sunKeyboard** resource is false, *xterm* recognizes function key modifiers which are parameters appended before the final character of the control sequence.

Code	Modifiers
2	**Shift**
3	**Alt**
4	**Shift + Alt**
5	**Control**
6	**Shift + Control**
7	**Alt + Control**
8	**Shift + Alt + Control**

For example, shift-F5 would be sent as CSI 1 5 ; 2 ~

VT220-Style Function Keys

However, *xterm* is most useful as a DEC VT102 or VT220 emulator. Set the **sunKeyboard** resource to true to force a Sun/PC keyboard to act like a VT220 keyboard.

The VT102/VT220 application keypad transmits unique escape sequences in application mode, which are distinct from the cursor and scrolling keypad:

Key	Numeric	Application

Space	**Space**	SS3 SP
Tab	**Tab**	SS3 I
Enter	**CR**	SS3 M
PF1	SS3 P	SS3 P
PF2	SS3 Q	SS3 Q
PF3	SS3 R	SS3 R
PF4	SS3 S	SS3 S
* (multiply)	*	SS3 j
+ (add)	+	SS3 k
, (comma)	,	SS3 l
- (minus)	-	SS3 m
. (period)	.	SS3 n
/ (divide)	/	SS3 o
0	**0**	SS3 p
1	**1**	SS3 q
2	**2**	SS3 r
3	**3**	SS3 s
4	**4**	SS3 t
5	**5**	SS3 u
6	**6**	SS3 v
7	**7**	SS3 w
8	**8**	SS3 x
9	**9**	SS3 y
= (equal)	=	SS3 X

The VT220 provides 8 additional function keys. With a Sun/PC keyboard, access these keys by Control/F1 for F13, etc.

Key	Escape Sequence
F13	CSI 2 5 ~
F14	CSI 2 6 ~
F15	CSI 2 8 ~
F16	CSI 2 9 ~
F17	CSI 3 1 ~
F18	CSI 3 2 ~
F19	CSI 3 3 ~
F20	CSI 3 4 ~

VT52-Style Function Keys

A VT52 does not have function keys, but it does have a numeric keypad and cursor keys. They differ from the other emulations by the prefix. Also, the cursor keys do not change:

Key	Normal/Application
Cursor Up	ESC A
Cursor Down	ESC B
Cursor Right	ESC C
Cursor Left	ESC D

The keypad is similar:

Key	Numeric	Application
Space	**Space**	ESC ? SP
Tab	**Tab**	ESC ? I
Enter	**CR**	ESC ? M

PF1	ESC **P**	ESC **P**
PF2	ESC **Q**	ESC **Q**
PF3	ESC **R**	ESC **R**
PF4	ESC **S**	ESC **S**
* (multiply)	*	ESC ? **j**
+ (add)	**+**	ESC ? **k**
, (comma)	**,**	ESC ? **l**
- (minus)	**-**	ESC ? **m**
. (period)	**.**	ESC ? **n**
/ (divide)	**/**	ESC ? **o**
0	**0**	ESC ? **p**
1	**1**	ESC ? **q**
2	**2**	ESC ? **r**
3	**3**	ESC ? **s**
4	**4**	ESC ? **t**
5	**5**	ESC ? **u**
6	**6**	ESC ? **v**
7	**7**	ESC ? **w**
8	**8**	ESC ? **x**
9	**9**	ESC ? **y**
= (equal)	**=**	ESC ? **X**

Sun-Style Function Keys

The *xterm* program provides support for Sun keyboards more directly, by a menu toggle that causes it to send Sun-style function key codes rather than VT220. Note, however, that the *sun* and *vt100* emulations are not really compatible. For example, their wrap-margin behavior differs.

Only function keys are altered; keypad and cursor keys are the same. The emulation responds identically. See the xterm-sun terminfo entry for details.

HP-Style Function Keys

Similarly, *xterm* can be compiled to support HP keyboards. See the xterm-hp terminfo entry for details.

Mouse Tracking

The VT widget can be set to send the mouse position and other information on button presses. These modes are typically used by editors and other full-screen applications that want to make use of the mouse.

There are six mutually exclusive modes. One is DEC Locator mode, enabled by the DECELR CSI P_s ; P_s ' z control sequence, and is not described here. The remaining five modes are each enabled (or disabled) by a different parameter in DECSET CSI ? P_m h or DECRST CSI ? P_m l control sequence.

Manifest constants for the parameter values are defined in **xcharmouse.h** as follows:

```
#define SET_X10_MOUSE              9
#define SET_VT200_MOUSE            1000
#define SET_VT200_HIGHLIGHT_MOUSE  1001
#define SET_BTN_EVENT_MOUSE        1002
#define SET_ANY_EVENT_MOUSE        1003
```

The motion reporting modes are strictly *xterm* extensions, and are not part of any standard, though they are analogous to the DEC VT200 DECELR locator reports.

Parameters (such as pointer position and button number) for all mouse tracking escape sequences generated by *xterm* encode numeric parameters in a single character as *value*+32. For example, ! specifies the value 1. The upper left character position on the terminal is denoted as 1,1.

X10 compatibility mode sends an escape sequence only on button press, encoding the location and the mouse button pressed. It is enabled by specifying parameter 9 to DECSET. On button press, *xterm* sends CSI M $C_b C_x C_y$ (6 characters). C_b is button−1. C_x and C_y are the x and y coordinates of the mouse when the button was pressed.

Normal tracking mode sends an escape sequence on both button press and release. Modifier key (shift, ctrl, meta) information is also sent. It is enabled by specifying parameter 1000 to DECSET. On button press or release, *xterm* sends CSI M $C_b C_x C_y$. The low two bits of C_b encode button information: 0=MB1 pressed, 1=MB2 pressed, 2=MB3 pressed, 3=release. The next three bits encode the modifiers which were down when the button was pressed and are added together: 4=Shift, 8=Meta, 16=Control. Note however that the shift and control bits are normally unavailable because *xterm* uses the control modifier with mouse for popup menus, and the shift modifier is used in the default translations for button events. The *Meta* modifier recognized by *xterm* is the *mod1* mask, and is not necessarily the "Meta" key (see *xmodmap*). C_x and C_y are the x and y coordinates of the mouse event, encoded as in X10 mode.

Wheel mice may return buttons 4 and 5. Those buttons are represented by adding 64 to the event code.

Mouse hilite tracking notifies a program of a button press, receives a range of lines from the program, highlights the region covered by the mouse within that range until button release, and then sends the program the release coordinates. It is enabled by specifying parameter 1001 to DECSET. Highlighting is performed only for button 1, though other button events can be received. **Warning:** use of this mode requires a cooperating program or it will hang *xterm.* On button press, the same information as for normal tracking is generated; *xterm* then waits for the program to send mouse tracking information. *All X events are ignored until the proper escape sequence is received from the pty:* CSI P_s ; P_s ; P_s ; P_s ; P_s T. The parameters are *func, startx, starty, firstrow,* and *lastrow. func* is non-zero to initiate hilite tracking and zero to abort. *startx* and *starty* give the starting x and y location for the highlighted region. The ending location tracks the mouse, but will never be above row *firstrow* and will always be above row *lastrow.* (The top of the screen is row 1.) When the button is released, *xterm* reports the ending position one of two ways: if the start and end coordinates are valid text locations: CSI t $C_x C_y$. If either coordinate is past the end of the line: CSI T $C_x C_y C_x C_y C_x C_y$. The parameters are *startx, starty, endx, endy, mousex,* and *mousey. startx, starty, endx,* and *endy* give the starting and ending character positions of the region. *mousex* and *mousey* give the location of the mouse at button up, which may not be over a character.

Button-event tracking is essentially the same as normal tracking, but *xterm* also reports button-motion events. Motion events are reported only if the mouse pointer has moved to a different character cell. It is enabled by specifying parameter 1002 to DECSET. On button press or release, *xterm* sends the same codes used by normal tracking mode. On button-motion events, *xterm* adds 32 to the event code (the third character, C_b). The other bits of the event code specify button and modifier keys as in normal mode. For example, motion into cell x,y with button 1 down is reported as CSI M @ $C_x C_y$. (@ = 32 + 0 (button 1) + 32 (motion indicator)). Similarly, motion with button 3 down is reported as CSI M B $C_x C_y$. (B = 32 + 2 (button 3) + 32 (motion indicator)).

Any-event mode is the same as button-event mode, except that all motion events are reported instead of just those that enter a new character cell. It is enabled by specifying 1003 to DECSET.

Tektronix 4014 Mode

Most of these sequences are standard Tektronix 4014 control sequences. Graph mode supports the 12-bit addressing of the Tektronix 4014. The major features missing are the write-through and defocused modes. This document does not describe the commands used in the various Tektronix plotting modes but does describe the commands to switch modes.

BEL Bell (Ctrl-G)

BS Backspace (Ctrl-H)

TAB Horizontal Tab (Ctrl-I)

LF	Line Feed or New Line (Ctrl-J)
VT	Cursor up (Ctrl-K)
FF	Form Feed or New Page (Ctrl-L)
CR	Carriage Return (Ctrl-M)
ESC ETX	Switch to VT100 Mode (ESC Ctrl-C)
ESC ENQ	Return Terminal Status (ESC Ctrl-E)
ESC FF	PAGE (Clear Screen) (ESC Ctrl-L)
ESC SO	Begin 4015 APL mode (ignored by *xterm*) (ESC Ctrl-N)
ESC SI	End 4015 APL mode (ignored by *xterm*) (ESC Ctrl-O)
ESC ETB	COPY (Save Tektronix Codes to file COPYyyyy-mm-dd.hh:mm:ss) (ESC Ctrl-W)
ESC CAN	Bypass Condition (ESC Ctrl-X)
ESC SUB	GIN mode (ESC Ctrl-Z)
ESC FS	Special Point Plot Mode (ESC Ctrl-\)
ESC 8	Select Large Character Set
ESC 9	Select #2 Character Set
ESC :	Select #3 Character Set
ESC ;	Select Small Character Set
OSC P_s ; P_t BEL	Set Text Parameters of VT window
	P_s = 0 → Change Icon Name and Window Title to P_t
	P_s = 1 → Change Icon Name to P_t
	P_s = 2 → Change Window Title to P_t
	P_s = 4 6 → Change Log File to P_t (normally disabled by a compile-time option)
ESC `	Normal Z Axis and Normal (solid) Vectors
ESC a	Normal Z Axis and Dotted Line Vectors
ESC b	Normal Z Axis and Dot-Dashed Vectors
ESC c	Normal Z Axis and Short-Dashed Vectors
ESC d	Normal Z Axis and Long-Dashed Vectors
ESC h	Defocused Z Axis and Normal (solid) Vectors
ESC i	Defocused Z Axis and Dotted Line Vectors
ESC j	Defocused Z Axis and Dot-Dashed Vectors
ESC k	Defocused Z Axis and Short-Dashed Vectors
ESC l	Defocused Z Axis and Long-Dashed Vectors
ESC p	Write-Thru Mode and Normal (solid) Vectors
ESC q	Write-Thru Mode and Dotted Line Vectors
ESC r	Write-Thru Mode and Dot-Dashed Vectors
ESC s	Write-Thru Mode and Short-Dashed Vectors
ESC t	Write-Thru Mode and Long-Dashed Vectors
FS	Point Plot Mode (Ctrl-\)
GS	Graph Mode (Ctrl-])
RS	Incremental Plot Mode (Ctrl-^)

US	Alpha Mode (Ctrl-_)

VT52 Mode

Parameters for cursor movement are at the end of the ESC Y escape sequence. Each ordinate is encoded in a single character as *value*+32. For example, ! is 1. The screen coodinate system is 0-based.

ESC A	Cursor up.
ESC B	Cursor down.
ESC C	Cursor right.
ESC D	Cursor left.
ESC F	Enter graphics mode.
ESC G	Exit graphics mode.
ESC H	Move the cursor to the home position.
ESC I	Reverse line feed.
ESC J	Erase from the cursor to the end of the screen.
ESC K	Erase from the cursor to the end of the line.
ESC Y P_s P_s	Move the cursor to given row and column.
ESC Z	Identify
	→ ESC / Z ("I am a VT52.")
ESC =	Enter alternate keypad mode.
ESC >	Exit alternate keypad mode.
ESC <	Exit VT52 mode (Enter VT100 mode).