NAME

attroff, **wattroff**, **attron**, **wattron**, **attrset**, **wattrset**, **color_set**, **wcolor_set**, **standend**, **wstandend**, **standout**, **wstandout**, **attr_get**, **wattr_get**, **attr_off**, **wattr_off**, **attr_on**, **wattr_on**, **attr_set**, **wattr_set**, **chgat**, **wchgat**, **mvchgat**, **mvwchgat**, **PAIR_NUMBER** − **curses** character and window attribute control routines

SYNOPSIS

```
#include <curses.h>
int attroff(int attrs);
int wattroff(WINDOW *win, int attrs);
int attron(int attrs);
int wattron(WINDOW *win, int attrs);
int attrset(int attrs);
int wattrset(WINDOW *win, int attrs);
int color_set(short color_pair_number, void* opts);
int wcolor_set(WINDOW *win, short color_pair_number,
     void* opts);
int standend(void);
int wstandend(WINDOW *win);
int standout(void);
int wstandout(WINDOW *win);
int attr_get(attr_t *attrs, short *pair, void *opts);
int wattr_get(WINDOW *win, attr_t *attrs, short *pair,
      void *opts);
int attr_off(attr_t attrs, void *opts);
int wattr_off(WINDOW *win, attr_t attrs, void *opts);
int attr_on(attr_t attrs, void *opts);
int wattr_on(WINDOW *win, attr_t attrs, void *opts);
int attr_set(attr_t attrs, short pair, void *opts);
int wattr_set(WINDOW *win, attr_t attrs, short pair, void *opts);
int chgat(int n, attr_t attr, short color,
     const void *opts)
int wchgat(WINDOW *win, int n, attr_t attr,
     short color, const void *opts)
int mvchgat(int y, int x, int n, attr_t attr,
     short color, const void *opts)
int mvwchgat(WINDOW *win, int y, int x, int n,
     attr_t attr, short color, const void *opts)
```

DESCRIPTION

These routines manipulate the current attributes of the named window. The current attributes of a window apply to all characters that are written into the window with **waddch**, **waddstr** and **wprintw**. Attributes are a property of the character, and move with the character through any scrolling and insert/delete line/character operations. To the extent possible, they are displayed as appropriate modifications to the graphic rendition of characters put on the screen.

The routine **attrset** sets the current attributes of the given window to *attrs*. The routine **attroff** turns off the named attributes without turning any other attributes on or off. The routine **attron** turns on the named attributes without affecting any others. The routine **standout** is the same as **attron(A_STANDOUT)**. The routine **standend** is the same as **attrset(A_NORMAL)** or **attrset(0)**, that is, it turns off all attributes.

The **attrset** and related routines do not affect the attributes used when erasing portions of the window. See **bkgd**(3NCURSES) for functions which modify the attributes used for erasing and clearing.

The routine **color_set** sets the current color of the given window to the foreground/background combination described by the color_pair_number. The parameter opts is reserved for future use, applications must supply a null pointer.

The routine **wattr_get** returns the current attribute and color pair for the given window; **attr_get** returns the current attribute and color pair for **stdscr**. The remaining **attr_*** functions operate exactly like the corresponding **attr*** functions, except that they take arguments of type **attr_t** rather than **int**.

The routine **chgat** changes the attributes of a given number of characters starting at the current cursor location of **stdscr**. It does not update the cursor and does not perform wrapping. A character count of −1 or greater than the remaining window width means to change attributes all the way to the end of the current line. The **wchgat** function generalizes this to any window; the **mvwchgat** function does a cursor move before acting. In these functions, the color argument is a color-pair index (as in the first argument of *init_pair*, see **color**(3NCURSES)). The **opts** argument is not presently used, but is reserved for the future (leave it **NULL**).

Attributes

The following video attributes, defined in **<curses.h>**, can be passed to the routines **attron**, **attroff**, and **attrset**, or OR'd with the characters passed to **addch**.

A_NORMAL	Normal display (no highlight)
A_STANDOUT	Best highlighting mode of the terminal.
A_UNDERLINE	Underlining
A_REVERSE	Reverse video
A_BLINK	Blinking
A_DIM	Half bright
A_BOLD	Extra bright or bold
A_PROTECT	Protected mode
A_INVIS	Invisible or blank mode
A_ALTCHARSET	Alternate character set
A_CHARTEXT	Bit-mask to extract a character
COLOR_PAIR(*n***)**	Color-pair number *n*

The following macro is the reverse of **COLOR_PAIR(***n***)**:

PAIR_NUMBER(*attrs*) Returns the pair number associated with the **COLOR_PAIR(***n***)** attribute.

The return values of many of these routines are not meaningful (they are implemented as macro-expanded assignments and simply return their argument). The SVr4 manual page claims (falsely) that these routines always return **1**.

NOTES

Note that **attroff**, **wattroff**, **attron**, **wattron**, **attrset**, **wattrset**, **standend** and **standout** may be macros.

COLOR_PAIR values can only be OR'd with attributes if the pair number is less than 256. The alternate functions such as **color_set** can pass a color pair value directly. However, ncurses ABI 4 and 5 simply OR this value within the alternate functions. You must use ncurses ABI 6 to support more than 256 color pairs.

PORTABILITY

These functions are supported in the XSI Curses standard, Issue 4. The standard defined the dedicated type for highlights, **attr_t**, which is not defined in SVr4 curses. The functions taking **attr_t** arguments are not supported under SVr4.

The XSI Curses standard states that whether the traditional functions **attron**/**attroff**/**attrset** can manipulate attributes other than **A_BLINK**, **A_BOLD**, **A_DIM**, **A_REVERSE**, **A_STANDOUT**, or **A_UNDERLINE** is "unspecified". Under this implementation as well as SVr4 curses, these functions correctly manipulate all other highlights (specifically, **A_ALTCHARSET**, **A_PROTECT**, and **A_INVIS**).

XSI Curses added the new entry points, **attr_get**, **attr_on**, **attr_off**, **attr_set**, **wattr_on**, **wattr_off**, **wattr_get**, **wattr_set**. These are intended to work with a new series of highlight macros prefixed with **WA_**.

Older versions of this library did not force an update of the screen when changing the attributes. Use **touchwin** to force the screen to match the updated attributes.

WA_NORMAL	Normal display (no highlight)
WA_STANDOUT	Best highlighting mode of the terminal.
WA_UNDERLINE	Underlining
WA_REVERSE	Reverse video
WA_BLINK	Blinking
WA_DIM	Half bright
WA_BOLD	Extra bright or bold
WA_ALTCHARSET	Alternate character set

The XSI curses standard specifies that each pair of corresponding **A_** and **WA_**-using functions operates on the same current-highlight information.

The XSI standard extended conformance level adds new highlights **A_HORIZONTAL**, **A_LEFT**, **A_LOW**, **A_RIGHT**, **A_TOP**, **A_VERTICAL** (and corresponding **WA_** macros for each) which this implementation does not yet support.

RETURN VALUE

All routines return the integer **OK** on success, or **ERR** on failure.

X/Open does not define any error conditions.

This implementation returns an error if the window pointer is null. The **wcolor_set** function returns an error if the color pair parameter is outside the range 0..COLOR_PAIRS−1. This implementation also provides **getattrs** for compatibility with older versions of curses.

Functions with a "mv" prefix first perform a cursor movement using **wmove**, and return an error if the position is outside the window, or if the window pointer is null.

SEE ALSO

ncurses(3NCURSES), **addch**(3NCURSES), **addstr**(3NCURSES), **bkgd**(3NCURSES), **printw**(3NCURSES), **curses_variables**(3NCURSES)