

NAME

clearok, idcok, idlok, immedok, leaveok, scrolllok, setscrreg, wsetscreg – set *curses* output options

SYNOPSIS

```
#include <curses.h>

int clearok(WINDOW * win, bool bf);
void idcok(WINDOW * win, bool bf);
int idlok(WINDOW * win, bool bf);
void immedok(WINDOW * win, bool bf);
int leaveok(WINDOW * win, bool bf);
int scrolllok(WINDOW * win, bool bf);

int setscreg(int top, int bot);
int wsetscreg(WINDOW * win, int top, int bot);
```

DESCRIPTION

These functions configure properties of *curses* windows that affect their manner of output. Boolean-valued properties are initially **FALSE** except where noted. **endwin(3X)** resets any terminal modes corresponding to these properties; an application need not restore their initial values.

clearok

Setting *win*'s **clearok** property to **TRUE** causes the next **wrefresh** call on it to clear the terminal screen and redraw it entirely. This property is useful to restore the contents of the screen (perhaps because another process has written to the terminal), or in some cases to achieve a more pleasing visual effect. If *win* is **curscr** (see **curs_variables(3X)**), the next **wrefresh** call on *any* window causes the terminal screen to clear and redraw as above. **wrefresh** resets this property to **FALSE**.

idcok

(This property defaults **TRUE**.) Setting *win*'s **idcok** property to **FALSE** prevents *curses* from using the insert/delete character capabilities of terminal types possessing them according to the *terminfo* database.

idlok

Setting *win*'s **idlok** property to **TRUE** causes *curses* to consider using the insert/delete line capabilities of terminal types possessing them according to the *terminfo* database. Enable this option if the application explicitly requires these operations, as a full-screen text editor might; otherwise the results may be visually annoying to the user.

immedok

If **immedok** is called with **TRUE** as second argument, changes to the window image, such as those caused by **waddch**, **wclrbot**, or **wscrl**, automatically cause a call to **wrefresh**. Setting a window's **immedok** property may degrade performance considerably if writes are frequent. Calling **immedok** with **FALSE** as second argument restores the default behavior, deferring screen updates until a refresh is needed or explicitly directed by the application.

leaveok

Normally, *curses* leaves the hardware cursor at the library's cursor location of the window being refreshed. The **leaveok** option allows the cursor to be left wherever the update happens to leave it. It is useful for applications that do not employ a visible cursor, since it reduces the need for cursor motions.

scrolllok

The **scrolllok** option controls what happens when a window's cursor moves off the edge of the window or scrolling region, as a result of either (1) writing a newline anywhere on its bottom line, or (2) writing a character that advances the cursor to the last position on its bottom line. If disabled (*bf* is **FALSE**), *curses* leaves the cursor on the bottom line of the window. If enabled (*bf* is **TRUE**), *curses* scrolls the window up one line. (To get the physical scrolling effect on the terminal, the application must also enable **idlok**).

setscreg, wsetscreg

The **wsetscreg** and **setscreg** functions allow the application to set a software scrolling region in the specified window or **stdscr**, respectively. The *top* and *bot* parameters are the line numbers of the top and bottom margin of the scrolling region. If this option and **scrolllok** are enabled, an attempt to move off the bottom

margin line causes all lines in the scrolling region to scroll one line in the direction of the first line. Only the text of the window is scrolled. (This process has nothing to do with the scrolling region capability of the terminal, as found in the DEC VT100.) If **idlok** is enabled and the terminal has either a scrolling region or insert/delete line capability, they will probably be used by the output routines.

RETURN VALUE

The functions **setscrreg** and **wsetscrreg** return **OK** upon success and **ERR** upon failure. All other routines that return an integer always return **OK**.

In *ncurses*, these functions fail if

- the *curses* screen has not been initialized,
- (for functions taking a *WINDOW* pointer argument) *win* is a null pointer, or
- (for **setscrreg** and **wsetscrreg**) the function is passed arguments describing a scrolling region with limits that extend outside the window boundaries.

NOTES

clearok, **leaveok**, **scrollok**, **idcok**, and **setscrreg** may be implemented as macros.

Unlike the other functions described by this page, **setscrreg** does not accept a pointer-to-*WINDOW* parameter, but operates on **stdscr**. Use **wsetscrreg** to configure the scrolling region of a selected window.

Historically, applications used **idcok(FALSE)** to accommodate the “magic cookie” feature of some terminal types; see subsection “Highlighting, Underlining, and Visible Bells” of **terminfo(5)**. When updating a line, the presence of character cells with magic cookies in them made the *curses* library’s computations of characters to be rewritten inaccurate. A better solution is to indicate the **magic_cookie_glitch (xmc)** capability in the terminal’s type description.

immedok is useful for windows that are used as terminal emulators.

PORTABILITY

X/Open Curses Issue 4 describes these functions. It specifies no error conditions for them.

Some historic *curses* implementations, as an undocumented feature, did the equivalent of “**clearok(..., 1)**” when **touchwin(stdscr)** or **clear(stdscr)** were used. This trick does not work with *ncurses*.

Early System V *curses* implementations specified that with *scrollok* enabled, any window modification triggering a scroll also forced a physical refresh. X/Open Curses does not require this, and *ncurses* avoids doing so to better optimize vertical motions upon a *wrefresh*.

X/Open Curses does not mention that the cursor should be made invisible as a side effect of *leaveok*. SVr4 *curses* documentation notes this behavior, but the code neglects to implement it. Use **curs_set(3X)** to make the cursor invisible.

HISTORY

4BSD (1980) introduced *clearok*, *leaveok*, and *scrollok*.

SVr2 (1984) supplied *idlok*, *setscrreg*, and *wsetscrreg*.

SVr3.1 (1987) implemented *idcok* and *immedok*.

ncurses formerly treated *nl* and *nonl* as both input and output options, but no longer; see **curs_inopts(3X)**.

SEE ALSO

curses(3X), **curs_addch(3X)**, **curs_clear(3X)**, **curs_initscr(3X)**, **curs_refresh(3X)**, **curs_scroll(3X)**, **curs_variables(3X)**