

**NAME**

**clearok**, **idlok**, **idcok**, **immedok**, **leaveok**, **setscrreg**, **wsetscrreg**, **scrollok** - set *curses* output options

**SYNOPSIS**

```
#include <curses.h>
```

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

int setscrreg(int top, int bot);
int wsetscrreg(WINDOW *win, int top, int bot);
```

**DESCRIPTION**

These routines set options that change the style of output within **curses**. All options are initially **FALSE**, unless otherwise stated. It is not necessary to turn these options off before calling **endwin(3X)**.

**clearok**

If **clearok** is called with **TRUE** as argument, the next call to **wrefresh** with this window will clear the screen completely and redraw the entire screen from scratch. This is useful when the contents of the screen are uncertain, or in some cases for a more pleasing visual effect. If the *win* argument to **clearok** is the global variable **curscr**, the next call to **wrefresh** with any window causes the screen to be cleared and repainted from scratch.

**idlok**

If **idlok** is called with **TRUE** as second argument, **curses** considers using the hardware insert/delete line feature of terminals so equipped. Calling **idlok** with **FALSE** as second argument disables use of line insertion and deletion. This option should be enabled only if the application needs insert/delete line, for example, for a screen editor. It is disabled by default because insert/delete line tends to be visually annoying when used in applications where it is not really needed. If insert/delete line cannot be used, **curses** redraws the changed portions of all lines.

**idcok**

If **idcok** is called with **FALSE** as second argument, **curses** no longer considers using the hardware insert/delete character feature of terminals so equipped. Use of character insert/delete is enabled by default. Calling **idcok** with **TRUE** as second argument re-enables use of character insertion and

deletion.

### **immedok**

If **immedok** is called with **TRUE** as second argument, any change in the window image, such as the ones caused by **waddch**, **wclrtoobot**, **wscr1**, etc., automatically causes a call to **wrefresh**. However, it may degrade performance considerably, due to repeated calls to **wrefresh**. Calling **immedok** with **FALSE** as second argument restores the default behavior, i.e., deferring screen updates until a refresh is needed.

### **leaveok**

Normally, the hardware cursor is left at the location of the window cursor being refreshed. The **leaveok** option allows the cursor to be left wherever the update happens to leave it. It is useful for applications where the cursor is not used, since it reduces the need for cursor motions.

### **scrollok**

The **scrollok** option controls what happens when the cursor of a window is moved off the edge of the window or scrolling region, either as a result of a newline action on the bottom line, or typing the last character of the last line. If disabled, (*bf* is **FALSE**), the cursor is left on the bottom line. If enabled, (*bf* is **TRUE**), the window is scrolled up one line (Note that to get the physical scrolling effect on the terminal, it is also necessary to call **idlok**).

### **setscrreg, wsetscrreg**

The **setscrreg** and **wsetscrreg** routines allow the application programmer to set a software scrolling region in a window. The *top* and *bot* parameters are the line numbers of the top and bottom margin of the scrolling region. (Line 0 is the top line of the window.) If this option and **scrollok** 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. (Note that this has nothing to do with the use of a physical scrolling region capability in the terminal, like that in the 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**.

X/Open Curses does not specify any error conditions.

In this implementation,

- ⊕ those functions that have a window pointer will return an error if the window pointer is null

- ⊕ **wsetscrreg** returns an error if the scrolling region limits extend outside the window boundaries.

## NOTES

Note that **clearok**, **leaveok**, **scrollok**, **idcok**, and **setscrreg** may be macros.

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

## PORTABILITY

These functions are described in *X/Open Curses*, Issue 4.

Some historic curses implementations had, as an undocumented feature, the ability to do the equivalent of **clearok(..., 1)** by saying **touchwin(stdscr)** or **clear(stdscr)**. This will not work under *ncurses*.

Earlier 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 it to perform better vertical-motion optimization at **wrefresh** time.

*X/Open Curses* does not mention that the cursor should be made invisible as a side-effect of **leaveok**. SVr4 curses documentation does this, but the code does not. Use  **curs\_set** to make the cursor invisible.

## HISTORY

*ncurses* formerly treated **nl(3X)** and **nonl(3X)** 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)**