**NAME**

curs_outopts, clearok, idlok, idcok, immedok, leaveok, setscrreg, wsetscrreg, scrollok, nl, nonl – curses terminal output option control routines

**SYNOPSIS**

**cc** [ *flag* ... ] *file* ... **–lcurses** [ *library* .. ]

**#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 setscrreg(int** *top*, **int** *bot***);**

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

**int scrollok(WINDOW** **win*, **bool** *bf***);**

**int nl(void);**

**int nonl(void);**

**MT-LEVEL**

Unsafe

**DESCRIPTION**

These routines set options that deal with output within **curses**. All options are initially **FALSE**, unless otherwise stated. It is not necessary to turn these options off before calling **endwin( )**.

With the **clearok( )** routine, if enabled (*bf* is **TRUE**), 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.

With the **idlok( )** routine, if enabled (*bf* is **TRUE**), **curses** considers using the hardware insert/delete line feature of terminals so equipped. If disabled (*bf* is **FALSE) , curses** very seldom uses this feature. (The insert/delete character feature is always considered.) 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 isn't really needed. If insert/delete line cannot be used, **curses** redraws the changed portions of all lines.

With the **idcok( )** routine, if enabled (*bf* is **TRUE**), **curses** considers using the hardware insert/delete character feature of terminals so equipped. This is enabled by default.

With the **immedok( )** routine, if enabled (*bf* is **TRUE**), any change in the window image, such as the ones caused by **waddch( )**, **wclrtobot( )**, **wscrl( )**, etc., automatically cause a call to **wrefresh( )**. However, it may degrade the performance considerably, due to repeated calls to **wrefresh( )**. It is disabled by default. 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. If possible, the cursor is made invisible when this option is enabled.

The **setscrreg( )** and **wsetscrreg( )** routines allow the application programmer to set a software scrolling region in a window. *top* and *bot* 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 up one 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.)

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**), **wrefresh( )** is called on the window, and the physical terminal and window are scrolled up one line. (Note that in order to get the physical scrolling effect on the terminal, it is also necessary to call **idlok( )**.)

The **nl( )** and **nonl( )** routines control whether newline is translated into carriage return and linefeed on output, and whether return is translated into newline on input. Initially, the translations do occur. By disabling these translations using **nonl( )**, **curses** is able to make better use of the linefeed capability, resulting in faster cursor motion.

## RETURN VALUES

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

## SEE ALSO

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

## NOTES

The header **<curses.h>** automatically includes the headers **<stdio.h>** and **<unctrl.h>**.

Note that **clearok( )**, **leaveok( )**, **scrollok( )**, **idcok( )**, **nl( )**, **nonl( )**, and **setscrreg( )** may be macros.

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