NAME

use_default_colors, **assume_default_colors -** use terminal’s default colors

SYNOPSIS

#include <curses.h>

int use_default_colors(void);
int assume_default_colors(int *fg*, **int** *bg***);**

DESCRIPTION

The **use_default_colors** and **assume_default_colors** functions are extensions to the curses library. They are used with terminals that support ISO 6429 color, or equivalent. These terminals allow the application to reset color to an unspecified default value (e.g., with SGR 39 or SGR 49).

Applications that paint a colored background over the whole screen do not take advantage of SGR 39 and SGR 49. Some applications are designed to work with the default background, using colors only for text. For example, there are several implementations of the **ls** program which use colors to denote different file types or permissions. These "color ls" programs do not necessarily modify the background color, typically using only the **setaf** terminfo capability to set the foreground color. Full-screen applications that use default colors can achieve similar visual effects.

The first function, **use_default_colors** tells the curses library to assign terminal default foreground/background colors to color number -1. So init_pair(x,COLOR_RED,-1) will initialize pair x as red on default background and init_pair(x,-1,COLOR_BLUE) will initialize pair x as default foreground on blue.

The other, **assume_default_colors** is a refinement which tells which colors to paint for color pair 0. This function recognizes a special color number -1, which denotes the default terminal color.

The following are equivalent:

use_default_colors();
assume_default_colors(-1,-1);

These are ncurses extensions. For other curses implementations, color number -1 does not mean anything, just as for ncurses before a successful call of **use_default_colors** or **assume_default_colors**.

Other curses implementations do not allow an application to modify color pair 0. They assume that the background is COLOR_BLACK, but do not ensure that the color pair 0 is painted to match the assumption. If your application does not use either **use_default_colors** or **assume_default_colors** ncurses will paint a white foreground (text) with black background for color pair 0.

RETURN VALUE

These functions return the integer **ERR** upon failure and **OK** on success. They will fail if either the terminal does not support the **orig_pair** or **orig_colors** capability. If the **initialize_pair** capability is not found, this causes an error as well.

NOTES

Associated with this extension, the **init_pair** function accepts negative arguments to specify default foreground or background colors.

The **use_default_colors** function was added to support *ded*. This is a full-screen application which uses curses to manage only part of the screen. The bottom portion of the screen, which is of adjustable size, is left uncolored to display the results from shell commands. The top portion of the screen colors filenames using a scheme like the "color ls" programs. Attempting to manage the background color of the screen for this application would give unsatisfactory results for a variety of reasons. This extension was devised after noting that color xterm (and similar programs) provides a background color which does not necessarily correspond to any of the ANSI colors. While a special terminfo entry could be constructed using nine colors, there was no mechanism provided within curses to account for the related **orig_pair** and **back_color_erase** capabilities.

The **assume_default_colors** function was added to solve a different problem: support for applications which would use environment variables and other configuration to bypass curses' notion of the terminal's default colors, setting specific values.

PORTABILITY

These routines are specific to ncurses. They were not supported on Version 7, BSD or System V implementations. It is recommended that any code depending on them be conditioned using NCURSES_VERSION.

SEE ALSO

ded(1), **curs_color**(3X).

AUTHOR

Thomas Dickey (from an analysis of the requirements for color xterm for XFree86 3.1.2C, February 1996).