NAME

> **bkgdset**, **wbkgdset**, **bkgd**, **wbkgd**, **getbkgd** - **curses** window background manipulation routines

SYNOPSIS

> **#include <curses.h>**
>
> **void bkgdset(chtype** *ch***);**
> **void wbkgdset(WINDOW ****win, chtype ch***);**
>
> **int bkgd(chtype** *ch***);**
> **int wbkgd(WINDOW ****win***, chtype** *ch***);**
>
> **chtype getbkgd(WINDOW ****win***);**

DESCRIPTION

> **bkgdset**
>
> The **bkgdset** and **wbkgdset** routines manipulate the background of the named window. The window background is a **chtype** consisting of any combination of attributes (i.e., rendition) and a character. The attribute part of the background is combined (OR'ed) with all non-blank characters that are written into the window with **waddch**. Both the character and attribute parts of the background are combined with the blank characters. The background becomes a property of the character and moves with the character through any scrolling and insert/delete line/character operations.
>
> To the extent possible on a particular terminal, the attribute part of the background is displayed as the graphic rendition of the character put on the screen.
>
> **bkgd**
>
> The **bkgd** and **wbkgd** functions set the background property of the current or specified window and then apply this setting to every character position in that window. According to X/Open Curses, it should do this:
>
> ⊕   The rendition of every character on the screen is changed to the new background rendition.
>
> ⊕   Wherever the former background character appears, it is changed to the new background character.
>
> Neither X/Open Curses nor the SVr4 manual pages give details about the way the rendition of characters on the screen is updated when **bkgd** or **wbkgd** is used to change the background character.
>
> This implementation, like SVr4 curses, does not store the background and window attribute

contributions to each cell separately. It updates the rendition by comparing the character, non-color attributes and colors contained in the background. For each cell in the window, whether or not it is blank:

- The library first compares the *character*, and if it matches the current character part of the background, it replaces that with the new background character.

- The library then checks if the cell uses color, i.e., its color pair value is nonzero. If not, it simply replaces the attributes and color pair in the cell with those from the new background character.

- If the cell uses color, and that matches the color in the current background, the library removes attributes which may have come from the current background and adds attributes from the new background. It finishes by setting the cell to use the color from the new background.

- If the cell uses color, and that does not match the color in the current background, the library updates only the non-color attributes, first removing those which may have come from the current background, and then adding attributes from the new background.

If the background's character value is zero, a space is assumed.

If the terminal does not support color, or if color has not been started with **start_color**, the new background character's color attribute will be ignored.

getbkgd
The **getbkgd** function returns the given window's current background character/attribute pair.

RETURN VALUE
These functions are described in the XSI Curses standard, Issue 4. It specifies that **bkgd** and **wbkgd** return **ERR** on failure, but gives no failure conditions.

The routines **bkgd** and **wbkgd** return the integer **OK**, unless the library has not been initialized.

In contrast, the SVr4.0 manual says **bkgd** and **wbkgd** may return **OK** "or a non-negative integer if **immedok** is set", which refers to the return value from **wrefresh** (used to implement the immediate repainting). The SVr4 curses **wrefresh** returns the number of characters written to the screen during the refresh. This implementation does not do that.

NOTES
Note that **bkgdset** and **bkgd** may be macros.

X/Open Curses mentions that the character part of the background must be a single-byte value. This implementation, like SVr4, checks to ensure that, and will reuse the old background character if the check fails.

PORTABILITY

These functions are described in the XSI Curses standard, Issue 4 (X/Open Curses).

SEE ALSO

curses(3X), **curs_addch**(3X), **curs_attr**(3X), **curs_outopts**(3X)