`!curses.panel` --- A panel stack extension for curses

- Functions
- Panel objects

curses.panel

Panels are windows with the added feature of depth, so they can be stacked on top of each other, and only the visible portions of each window will be displayed. Panels can be added, moved up or down in the stack, and removed.

Functions

The module `!curses.panel` defines the following exception:

error

Exception raised when a curses panel library function returns an error.

The module `!curses.panel` defines the following functions:

bottom_panel()

Returns the bottom panel in the panel stack.

new_panel(win)

Returns a panel object, associating it with the given window *win* and placing the new panel on top of the panel stack. Be aware that you need to keep the returned panel object referenced explicitly. If you don't, the panel object is garbage collected and removed from the panel stack.

top_panel()

Returns the top panel in the panel stack.

update_panels()

Updates the virtual screen after changes in the panel stack. This does not call `curses.doupdate`, so you'll have to do this yourself.

Panel objects

Panel objects, as returned by `new_panel` above, are windows with a stacking order. There's always a window associated with a panel which determines the content, while the panel methods are responsible for the window's depth in the panel stack.

Panel objects have the following methods:

panel.above()

Returns the panel above the current panel.

panel.below()

Returns the panel below the current panel.

panel.bottom()

Push the panel to the bottom of the stack.

panel.hidden()

Returns `True` if the panel is hidden (not visible), `False` otherwise.

panel.hide()

Hide the panel. This does not delete the object, it just makes the window on screen invisible.

panel.move(y, x)

Move the panel to the screen coordinates `(y, x)`.

panel.replace(win)

Change the window associated with the panel to the window *win*. Raise `curses.panel.error` if *win* has been detached from its screen by `screen.close() <curses.screen.close>`.

panel.set_userptr(obj)

Set the panel's user pointer to *obj*. This is used to associate an arbitrary piece of data with the panel, and can be any Python object.

panel.show()

Display the panel (which might have been hidden), placing it on top of the panel stack.

panel.top()

Push panel to the top of the stack.

panel.userptr()

Returns the user pointer for the panel. This might be any Python object.

panel.window()

Returns the window object associated with the panel.

□□□ #2
□ □□□ □□□ 2026-10-31 17:07:24 UTC
□ □□□ □□□ 2026-10-31 17:07:26 UTC