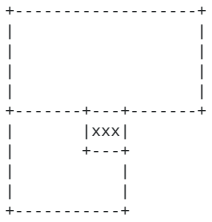
X11 struts geometry

Ask Question

Can somebody explain (or link to explanation) of how X11 struts are constructed? From the only description that I was able to find here it is not clear to me what the twelve values in _NET_WM_STRUT_PARTIAL property represent. For example, in the situation below where I have two Xinerama displays aligned on the left edge, how would I define the strut (marked xxx)? I suppose that the origin of the coordinates is at the top left hand corner.



x11 window-managers

asked Jan 27 '14 at 15:30

John Newman
83 1 6

1 Answer

This page explains `_NET_WM_STRUT_PARTIAL` :

```
_NET_WM_STRUT_PARTIAL, left, right, top, bottom, left_start_y, left_end_y,
right_start_y, right_end_y, top_start_x, top_end_x, bottom_start_x,
bottom_end_x,CARDINAL[12]/32
```

> [...] For example, for a panel-style Client appearing at the bottom of the screen, 50 pixels tall, and occupying the space from 200-600 pixels from the left of the screen edge would set a bottom strut of 50, and set bottom_start_x to 200 and bottom_end_x to 600. **Another example is a panel on a screen using the Xinerama extension.** Assume that the set up uses two monitors, one running at 1280x1024 and the other to the right running at 1024x768, with the top edge of the two physical displays aligned. If the panel wants to fill the entire bottom edge of the smaller display with a panel 50 pixels tall, it should set a bottom strut of 306, with bottom_start_x of 1280, and bottom_end_x of 2303. Note that the strut is relative to the screen edge, and not the edge of the xinerama monitor.

(my bold face).

Now, how does this work? Think of it as a feature which is triggered by a non-0 value in the first 4 integers. So if you want to reserve space at the bottom, you set `left, right, top` to 0 and `bottom` to 50.

The `*_start_x` and `*_end_x` pairs then define the size of the reserved area along the side of the screen.

In your example, you want to reserve space at the right side of the screen. If you main area is 2000 pixel and the smaller screen is 1200 pixel wide and the area should be 150 pixel, then you need `bottom = 2000 - 1200 + 150 = 950` (the virtual screen in this setup is 2000 pixels wide everywhere and so you need to offset the value with the difference of the widths of the two real monitors).

`right_start_y` == height of upper monitor.

Aaron Digulla
211k 61 383 594

that's exactly the part I did not understand. If I assume that the coordinate system starts at the top left hand corner, than the bottom number does not add up. And how should I set the other values (left, right, top, etc.)? – John Newman Jan 27 '14 at 15:43

I added an example using your setup. – Aaron Digulla Jan 27 '14 at 15:51

ok, thanks, it starts to be more clear. What is the difference between bottom and top value? When would I set top value to non-zero? – John Newman Jan 27 '14 at 16:03

1 If you want an Apple-like menu bar: top=14, top_start_x=0, top_end_x = width of screen, everything else is 0. – Aaron Digulla Jan 27 '14 at 16:06

1 No. `bottom` uses the two `bottom_*` attributes and nothing else. `right_start_y` with `right=0` will have no effect. – Aaron Digulla Jan 27 '14 at 16:13