Subject: GET_DRAW_VIEW question revisited
Posted by Daniel Peduzzi on Fri, 16 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

A few weeks ago, I posted a message concerning machine-specific discrepancies which were being returned by the GET_DRAW_VIEW keyword of WIDGET_CONTROL. I think it was David Fanning who requested that I post a follow-up, if I ever received an answer.

I was interested in retrieving the viewport position for a scrolling draw widget via the GET_DRAW_VIEW keyword to WIDGET_CONTROL. This seems to work well for the case when the viewport size equals the size of the draw widget, but for all other cases when the viewport size is smaller than the actual widget size, I get the same non-zero offset of 19 (on an Ultrabook.)

As a work-around, I tried to calculate and compensate for the viewport y-offset, but I found that although the offset would help out at the top position of the scrollbar, there would then be a "gap" at the bottom of the widget (that is, the y_position + y_viewport_size would fall short of the ysize of the widget. This gap would equal the offset for which I was attempting to compensate at the top of the widget in the first place.)

After a few rounds of Email with RSI support, I received this final response. The bottom line: it's a bug.

=== ===
There was a bug logged on this issue a couple of years ago.
Supposedely it was resolved but now we're finding there are still
some problems. In particular, I find the x and y values returned in
get_draw_view depends on whether there is x and/or y scroll bars.

On the Sun if x_scroll lt xsize+2 then instead of a gap it's
like xsize and ysize are increased by 2. That is, the x and y values
returned by get_draw_view range from 0 to xsize+2 - x_scroll and 0 to
ysize+2 - y_scroll. On the other hand, if x_scroll ge xsize+2 (so there's no x scroll bar)
then the y-value returned by get_draw_view ranges between 0 and ysize - y_scroll - 21.
This latter situation is what we've been discussing in our emails, I think.

The problem seems to be symmetric in that x and y can be interchanged and
the same situation holds. The problem manifests differently on the Windows OS.

So in the end I'll be logging a bug, (probably to the dismay of the
developer who thought they had this fixed.)

=== ===

Fortunately, for what I was trying to do, this isn't a real show-stopper...just a little annoying.

Dan Peduzzi
peduzzi@mediaone.net