Subject: ScrollBarSize() - Is it (possibly) a bug?
Posted by Lance on Mon, 26 Sep 2022 23:35:08 GMT
View Forum Message <> Reply to Message

I was looking for some member function in ScrollBar to retieve the scrollbar size, or expose the const Style *style
which can in turn access the barsize value, but to no avail. Until I find this function

```
inline int ScrollBarSize()              { return ScrollBar::StyleDefault().barsize; }//!!
```

A search in the Upp src indicates many libraries make use of this function, including in HeaderCtrl

```
void HeaderCtrl::FrameLayout(Rect& r)
{
 LayoutFrameTop(r, this, invisible ? 0 : GetHeight());
 LayoutFrameBottom(r, &sb, sb.IsShown() ? ScrollBarSize() : 0);
}

void HeaderCtrl::FrameAddSize(Size& sz)
{
 if(!invisible)
  sz.cy += GetHeight();
 if(sb.IsVisible())
  sz.cy += ScrollBarSize();
}
```

Does this imply that a customized ScrollBar Style shall not disagree on barsize with the default ScrollBar Style? Is this restriction necessary? or shall we expose int ScrollBar::ScrollBarSize()const(elevate it to a public member function from a private one as it is now)?

Subject: Re: ScrollBarSize() - Is it (possibly) a bug?
Posted by mirek on Sat, 17 Dec 2022 13:15:23 GMT
View Forum Message <> Reply to Message

Lance wrote on Tue, 27 September 2022 01:35I was looking for some member function in ScrollBar to retieve the scrollbar size, or expose the const Style *style
which can in turn access the barsize value, but to no avail. Until I find this function

```
inline int ScrollBarSize()              { return ScrollBar::StyleDefault().barsize; }//!!
```

A search in the Upp src indicates many libraries make use of this function, including in HeaderCtrl

```
void HeaderCtrl::FrameLayout(Rect& r)
{
 LayoutFrameTop(r, this, invisible ? 0 : GetHeight());
 LayoutFrameBottom(r, &sb, sb.IsShown() ? ScrollBarSize() : 0);
}

void HeaderCtrl::FrameAddSize(Size& sz)
{
 if(!invisible)
  sz.cy += GetHeight();
 if(sb.IsVisible())
  sz.cy += ScrollBarSize();
}
```

Does this imply that a customized ScrollBar Style shall not disagree on barsize with the default ScrollBar Style? Is this restriction necessary? or shall we expose int ScrollBar::ScrollBarSize()const(elevate it to a public member function from a private one as it is now)?

What a nice grey corner have you sumpled upon...

Well, I would say, for now it means that if you use ScrollBar::SetStyle, you cannot expect your styled ScollBar be well integrated is some other widgets unless scrollbarsize is the same...

In practice, in over 15 years I do not remember single instance of custom Scrollbar style, so while this should be fixed, it is really low priority.

Subject: Re: ScrollBarSize() - Is it (possibly) a bug?
Posted by Lance on Sun, 18 Dec 2022 03:32:27 GMT
View Forum Message <> Reply to Message

Thank you, Mirek!

It so happened that a widget I wrote used both vertical ScrollBar and horizontal ScrollBar with both scrollbars set to autohide. An edge case arose that if I set the page width, page height, total width, total height programmatically and I set the page width=total width = width of the view, and when the total height is (maybe slightly) greater than view height, an oscillation kind of events happened: a change in the visibility of one scrollbar causes the change in the visibility of the other scrollbar, the GUI keeps flickering and no user input can be processed.

I have lost my old code, am not sure if I remember what happened exactly, and am not sure if I can still reproduce the event, but I swear it happened to me.

To remedy that, I decided to disable native ScrollBar autohide behaviour and write the autohide logic myself. For that purpose, I need to query the ScrollBarSize(), ideally not in a hackish way.

Currently I use the exposed global function but are feeling unsure/uncomfortable