Subject: Minor bug: TopWindow::GetStdSize()
Posted by mrjt on Wed, 03 Oct 2007 12:25:41 GMT
View Forum Message <> Reply to Message

Ctrl::GetStdSize() returns GetMinSize().cx*10 (A recent change?), but TopWindow and ParentCtrl should return GetMinSize unchanged.

The addition of a SetStdSize would also be useful, but since this would require the addition of a member I will leave it to your discretion.

Also:
Image HorzFadeOut(int cx, int cy, Color color); (ImageOp.h)
Is missing it's body. Presumably should call HorzFadeOut(Size sz, Color color).

Cheers,
James

Subject: Re: Minor bug: TopWindow::GetStdSize()
Posted by mrjt on Fri, 05 Oct 2007 12:54:23 GMT
View Forum Message <> Reply to Message

Also...

It would very helpful if ParentCtrl supported SetMinSize properly (ie. not with an empty template like other controls). Otherwise there is no way of telling how large a layout is.

```
struct ParentCtrl : public Ctrl {
 virtual Rect   GetVoidRect();

 ParentCtrl();

 virtual void SetMinSize(Size sz) { minsize = sz; }
 virtual Size GetMinSize() const  { return minsize; }
private:
 Size         minsize;
};
```

All of this stuff may seem somewhat trivial, but one of the things that is difficult in Upp is creating gerneric container widgets of any sort, simply because there is often no way of guessing how large a control should be in a layout.

With a few small changes this could be made much easier.

Cheers,

James

Subject: Re: Minor bug: TopWindow::GetStdSize()
Posted by mirek on Sun, 07 Oct 2007 17:29:01 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 03 October 2007 08:25Ctrl::GetStdSize() returns GetMinSize().cx*10 (A recent change?), but TopWindow and ParentCtrl should return GetMinSize unchanged.

The addition of a SetStdSize would also be useful, but since this would require the addition of a member I will leave it to your discretion.

Also:
Image HorzFadeOut(int cx, int cy, Color color); (ImageOp.h)
Is missing it's body. Presumably should call HorzFadeOut(Size sz, Color color).

Cheers,
James

Thanks, fixed.

Subject: Re: Minor bug: TopWindow::GetStdSize()
Posted by mirek on Sun, 07 Oct 2007 17:30:26 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 03 October 2007 08:25Ctrl::GetStdSize() returns GetMinSize().cx*10

Actually, not really recent, but stupid idea nevertheless.

I hope to be able to replace this to by simple GetMinSize() call, however I have to investigate carefuly the extent of the code that would be broken by such change...

Mirek

Subject: Re: Minor bug: TopWindow::GetStdSize()
Posted by mirek on Sun, 07 Oct 2007 17:33:21 GMT
View Forum Message <> Reply to Message

mrjt wrote on Fri, 05 October 2007 08:54Also...

It would very helpful if ParentCtrl supported SetMinSize properly (ie. not with an empty template like other controls). Otherwise there is no way of telling how large a layout is.

```
struct ParentCtrl : public Ctrl {
 virtual Rect   GetVoidRect();

 ParentCtrl();

 virtual void SetMinSize(Size sz) { minsize = sz; }
 virtual Size GetMinSize() const  { return minsize; }
private:
 Size         minsize;
};
```

All of this stuff may seem somewhat trivial, but one of the things that is difficult in Upp is creating gerneric container widgets of any sort, simply because there is often no way of guessing how large a control should be in a layout.

With a few small changes this could be made much easier.

Cheers,
James

I am not too sure, considering that you would need this for some layout management scheme, that such attribute is really a good idea.

I believe that such size should be in fact a result of the whole layout machinery somehow. This perhaps not really different from the size of the whole dialog and I believe that sizing the dialog based on the size of its children is the point.... so perhaps something similar should be used for ParentCtrl too...

Subject: Re: Minor bug: TopWindow::GetStdSize()
Posted by mrjt on Mon, 08 Oct 2007 08:53:49 GMT
View Forum Message <> Reply to Message

But how would this work? What you suggest (working out correct size based on children) implies some sort of additional layout logic that is not currently possible.

Personally I think SetMinSize makes perfect sense for ParentCtrl. Why should it be different from TopWindow?

You're the boss though

Edit: Incidentally, the expandable container frame I have just posted (here) is the reason for the previous posts. Currently it is necessary to explicitly give the size of the ParentCtrl when adding it

to the frame because there is no reliable way of guessing.

Subject: Re: Minor bug: TopWindow::GetStdSize()
Posted by mirek on Mon, 08 Oct 2007 16:59:43 GMT
View Forum Message <> Reply to Message

mrjt wrote on Mon, 08 October 2007 04:53
Personally I think SetMinSize makes perfect sense for ParentCtrl. Why should it be different from TopWindow?

The real purpose of SetMinSize of TopWindow to limit its resizing to the size of designed layout. I guess the whole point is to avoid designed layouts here

Mirek