Subject: Re: Know what you're using. Size of some common types.
Posted by Lance on Mon, 10 Jan 2022 20:43:41 GMT
View Forum Message <> Reply to Message

Buttons established a concept. If implemented with due care, it will arrive at same Ctrl size (eg, ScrollBar, SpinButtons) as when hardcoded (like the way I do ScrollBar). That being said, Buttons interface are not expected to save much work (than hard coded way), if any. Personally I will do without: I don't see the benefit here, except the visual clue that they are doing similar things.

I have performed a simple test on adding SpinButtons to EditField without resorting to a CtrlFrame, and it works like a charm. I know this is not conceptually right, but it will work in practice and save memory of a little over sizeof(Ctrl) for each EditWithSpin object than the Buttons way mirek is considering. The only functionality it lose is when there are multiple frames, you cannot choose to put the SpinButtons frame on anywhere except innermost, because it's not a real CtrlFrame.

The Test is very simple, start a CtrlLib project with a Main Window with a EditField and a EditInt64WithSpin. Now we start to play around with CtrlLib.

In <CtrlLib/EditCtrl.h, add a new virtual function to class EditField

```
class EditField : public Ctrl, private TextArrayOps {
public:
 virtual void  Layout();
 virtual void  Paint(Draw& draw);
 virtual void  LeftDown(Point p, dword keyflags);
        ... omitted
 virtual void   State(int);


        // newly introduced
 virtual Size   GetReducedSize()const{
              // EditField::GetReducedSize() should just return GetSize();
              // it's the WithSpin derived class that should override this
              // function to reserve room for SpinButtons.
              //
              // here we reduce it as if allocating space for SpinButtons
              Size sz = GetSize();
              sz.cx -= 30;
              if(sz.cx < 0)
                   sz.cx=0;
              return sz;
         }[/b]
...
```

In <CtrlLib/EditField.cpp>, search all occurence of "GetSize()", if it's not "GetSize().cy", and is not "GetParent()->GetSize()", repace it with "GetReducedSize()". Run the test program, notice the room for SpinButton has been allocated, and play with some input in the EditField, see it scroll horizontally properly when the text gets really long. Now we can be certain this route is feasible. And it will come out with more compact EditXXWithSpins than to retain Buttons in a CtrlFrame. I would say the amount of coding involved are quite similar in both ways.

I will otherwise leave EditField (except probably add a spin_visible bitfield member intended for EditXXwithSpin). Then override Paint, relevant mouse event virtual function in WithSpin<> template class. It's quite similar to what I did with ScrollBar. No impact will be felt by libary users except smaller EditXXWithSpin objects---class hierarchy and interfaces remain unchanged.