

---

Subject: Re: Know what you're using. Size of some common types.

Posted by [mirek](#) on Mon, 10 Jan 2022 00:07:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Lance wrote on Sun, 09 January 2022 20:36 If the intended modified Button will continue to be a Ctrl derivative so that it works seamlessly with existing code, the minimum size it will have is sizeof(Ctrl). In this particular situation, the Ctrl member variable LogPos (and maybe more candidates) can be done without, we can reuse it to put a Style \* st; and probably WhenPush event variable somewhere else. In any case, the minimum size will be sizeof(Ctrl). The savings won't be quite as good (significantly less than the case when buttons or even frames are ridden of), even though it indeed involves least work and least chance of surprise.

Another way is to define it something like

```
class FakeButton
{
public:
    void Paint(Draw& w, Rect& where, Button::Style& style, int
state/*normal,hot,presseed,disabled*/);

    Event<> WhenPush;
};
```

This way sizeof(FakeButton)==sizeof(Event<>)(==sizeof(void\*)) , but the effort involved in revising containing Ctrl (ScrollBar, EditWithSpin, DropList, etc)'s Paint,LeftDown,LeftRepeat, etc, will be quite similar to hardcoding each of the Ctrl's method that we need to change to allow the savings to take place.

This would be too long. What we need is class Buttons like this:

```
class Buttons : Ctrl {
    virtual int GetButtonCount() = 0;
    virtual Rect GetButtonRect(int i) = 0;
    virtual int ButtonMouseDown(Point p);
    virtual int ButtonMouseUp(Point p);
    virtual int ButtonMouseMove(Point p);
}
```