Subject: Re: Know what you're using. Size of some common types.
Posted by Lance on Fri, 07 Jan 2022 16:25:51 GMT
View Forum Message <> Reply to Message

mirek:

Thanks. The output from above test program works better with Excel like utility so that record keeping and comparison are easier.

While there will not be a fixed ratio between total Ctrl used and that of ScrollBar used, I have test run some examples to get a feel of a rough ratio.

Examples/AddressBook(Up to the mainwindow is open): Max Ctrl Used: 96, Max ScrollBar used: 11 (9:1)

Examples/HomeBudget(Up to the mainwindow is open): Max Ctrl Used: 277, Max ScrollBar used: 22 (13:1)

Reference/GridCtrlTest(Up to the maindown is open): Max Ctrl Used: 1011, Max ScrollBar used: 132 (8:1)

UppSrc/ide(open a blank CtrlLib application): Max Ctrl Used: 802, Max ScrollBar used: 217 (4:1);

Again ide, but this time open UppSrc/ide, and in it, click the very last file ide.lay: Max Ctrl Used: 22001, Max ScrollBar used: 2181 (10: 1)

Considering the absolute and percentage saving we derived from the new implementation of ScrollBar (well, only an insignificant part of it to be more precise), accepting new ScrollBar would be as beneficial as compacting Ctrl, if not more: mirek mentioned in another discussion that he could replace a String with a const char *, resulting in additional saving of 8 bytes on 64 bit platforms and 12 bytes on 32 bit ones. Combining with that derived from rearranging member variables to minimize padding, we end up with 16 bytes each on both 32 and 64 bit platforms. That's about it if we don't want to lose any functionalities.

ScrollBar is pretty isolated: I don't think many people will need to derive from it. As long as we maintain the user interface stable/untouched, and test it on different platforms/settings, it should pose very low risk of messing up things (to replace it). These are all gain at no cost: by the way, new ScrollBar results in smaller executable too. Now my question is: Why not? :p

---
PS: Even if you derive from ScrollBar, I don't think you will be affected: the functions/member variables changed are all private (as far as I can remember).