Subject: Re: A problem with UPP application scale . . . UPP exonerated, the rest of the story.
Posted by mirek on Wed, 25 Feb 2009 09:58:18 GMT
View Forum Message <> Reply to Message

jlfranks wrote on Thu, 19 February 2009 13:41Our UPP application has been growing for over a year.
We have over 75 screens, custom UPP widget library
designed for touch screen access, virtual keyboard for editing, etc.

The problem is the header and .lay files. Any modification
anywhere causes the world to recompile. Just adding white
space to a header file, or changing a label content (.lay)
causes a long compilation on a dual core machine with 2 GByte
RAM running openSuSE 10.2

Well, bad news first, there is a price to be paid for U++ design approach. The design decision is to reduce code complexity via close layout -> class relationsthip and strictly avoiding all kinds of unnecessarry pointers.

Then, to address the recompile problem, we have rather tried to actually make recompile FAST. Using BLITZ, you get compiled 8-16 files for a time you get compiled single one in large project, as a result of changing single label. BLITZ really makes "recompile because of header change" problem next to non-existant.

However, this really is not the end of the story. If you really wish to or need to, you can still keep .lay separated by kind of PIMPL, e.g.:

header:

```
class MyDialog {
    struct MyDialogImp;
    One<MyDialogImp> dlg;

    MyDialog() { dlg.Create(); }
};
```

.cpp :

```
#define  LAYOUTFILE "MyDialogLayout.lay"
#include <CtrlCore/lay.h>

class MyDialogImp : WithMyDialogLayout<TopWindow> {
};
```

or even:

header:

```
class MyDialog : TopWindow {
    struct MyDialogImp;
    One<MyDialogImp> dlg;

    MyDialog() { dlg.Create(); Add(dlg->SizePos()); }
};
```

.cpp :

```
#define  LAYOUTFILE "MyDialogLayout.lay"
#include <CtrlCore/lay.h>

class MyDialogImp : WithMyDialogLayout<ParentCtrl> {
};
```

I hope you got the idea... There are countless variations of this approach...

Personally, I would not do this, because you are sacrificing human work for compile time.

BTW, how many actual lines has the app? 75 screens ain't that many. My computer does full debug mode rebuild of theide (200000 lines of .c/.cpp and 50000 lines of .h/.lay) in about 25 second.

I suspect you are not using BLITZ (you are on ARM, right?), so maybe your first remedy would be to try BLITZ after all. Differences are DRAMATIC!

Speaking about it:

1) I would be interested to hear how well Painter works on ARM, given that it is using a lot of FP.

2) What is the status of your U++ sources? I believe you had to change quite a lot to make it work with your platform, are not you interested in adding some patches in future U++ to make your work easier?

Mirek