Subject: Separate Info Window
Posted by roger on Thu, 06 Apr 2017 21:34:48 GMT

I'm looking for a way to display a body of text in a separate window that won't be modal (while I do other stuff in the main window).

Exclamation() would work great if it wasn't modal.
The Multiple main windows example in the GUI tutorial might also work, but it doesn't seem to do embedded newlines (e.g., "line 1\nline2\nline3\n...");
(Also could not get DocEdit or TextCtrl to work for this, and tried various flavors of Prompt() from CtrlLib (with dlg.Run(false), dlg.Execute(), but they all seemed modal in performance, and dlg.OpenMain() and dlg.Open() threw Assertion failed).

Any suggestions would be appreciated. This shouldn't be this hard.

Subject: Re: Separate Info Window
Posted by roger on Fri, 07 Apr 2017 15:36:59 GMT

I finally got this working using something like:

in layout file:
LAYOUT (SeparateInfoWindowLayout, 300, 100)
  ITEM(DocEdit, content, HSizePosZ(10,20).VSizePosZ(30,40))
END_LAYOUT

and in C++ code:

#include "Layout.h"

class SeparateInfoWindow : public WithSeparateInfoWindowLayout<PlacementAware>
{
  typedef SeparateInfoWindow CLASNAME;
public:
  SeparateInfoWindow();
  void Initialize();
private:
  void OnCancelAction();
} ;

SeparateInfoWindow::SeparateInfoWindow()
{
  CtrlLayout(*this);
  Sizeable(true);
  MinimizeBox(true);

```
  MaximizeBox(true);
}

void SeparateInfoWindow::Initialize(){}

SeparateInfoWindow::OnCancelAction() { Close(); }
```

Still later in C++ code:

```
SeparateInfoWindow siw;

if(siw.IsOpen()); else siw.Sizeable().OpenMain();

siw.content.Set("strings with embedded \\n's\n\nand they wrapped horizontally ........\n\nand the
DocEdit sprouted a vertical scrollbar when needed! :))");
siw.SetReadOnly();


// (sorry about any typos, I copied this manually from the development system - which is not
connected.)
```