
Subject: Interesting....

Posted by [mirek](#) on Tue, 21 Mar 2006 20:41:58 GMT

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

<http://www.crazyontap.com/topic.php?TopicId=2141&Posts=7>

Quote:

An inability to generate GUIs on the fly, dynamically, is one of the suggested limitations of Ultimate++. I don't know if it's true or not.

I guess somebody (well, ok, it will be me) should create an example of dialog generated out of XML....

Mirek

Subject: Re: Interesting....

Posted by [mirek](#) on Tue, 21 Mar 2006 21:09:25 GMT

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

```
#include <CtrlLib/CtrlLib.h>
```

```
struct XmlDlg : public TopWindow {  
    Array<Label>          label;  
    ArrayMap<String, EditString> edit;  
    ArrayMap<String, Option>  option;
```

```
    bool Load(const char *xml);  
};
```

```
bool XmlDlg::Load(const char *xml)  
{  
    int y = 10;  
    try {  
        XmlParser p(xml);  
        while(!p.IsTag())  
            p.Skip();  
        p.PassTag("dialog");  
        int linecy = Draw::GetStdFontCy() + 4;  
        while(!p.End())  
            if(p.TagE("option")) {  
                Add(option.Add(p["id"]).SetLabel(p["label"]).TopPos(y, linecy).LeftPos(10, 100));  
                y += linecy + 4;  
            }  
        else  
            if(p.TagE("edit")) {
```

```

    Add(label.Add()).SetLabel(p["label"]).TopPos(y, linecy).LeftPos(10, 30));
    Add(edit.Add(p["id"]).TopPos(y, linecy).LeftPos(40, 60));
    y += linecy + 4;
}
else
    p.Skip();
}
catch(XmlError e) {
    Exclamation("XML error: " + e);
    return false;
}
SetRect(0, 0, 110, y + 10);
return true;
}

```

```

GUI_APP_MAIN
{
    XmlDlg dlg;
    if(!dlg.Load(LoadFile(GetDataFile("dialog.xml"))))
        return;
    dlg.Run();
    int q = dlg.edit.Find("E1");
    if(q >= 0)
        PromptOK("E1 value: " + DeQtf(AsString(~dlg.edit[q])));
}

```

```

<?xml version="1.0"?>
<dialog>
  <option id="O1" label="Option"/>
  <option id="O2" label="Another option"/>
  <edit id="E1" label="Text"/>
  <edit id="E2" label="Text2"/>
</dialog>

```

Subject: Re: Interesting....
 Posted by [unodgs](#) on Wed, 22 Mar 2006 07:45:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Very good example Mirek!
 Personally I don't like xml (html, and any kind of unreadable text format.. , but most of people should like it because it follows the market trends.. (xml everywhere..)

Subject: Re: Interesting....

Posted by [mirek](#) on Wed, 22 Mar 2006 09:09:11 GMT

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

unodgs wrote on Wed, 22 March 2006 02:45Very good example Mirek!

Personally I don't like xml (html, and any kind of unreadable text format.. , but most of people should like it because it follows the market trends.. (xml everywhere..)

Actually, this is more about creating layout dynamically...

Note there are still no "new"s and "delete"s

Mirek

Subject: Re: Interesting....

Posted by [hojtsy](#) on Wed, 22 Mar 2006 12:36:00 GMT

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

I am afraid this will not satisfy the critics. It is still just an example supporting only two control types, and very limited set of properties. And you are unable to place the controls to horizontal or vertical groups of controls. The window width, and control sizes are hardcoded. It shows that dinamic generation can be done from client code, but only if the client goes and implements all of it - it's not a feature ready to be used. (Situation is simmlar with HtmlBrowser control)

Subject: Re: Interesting....

Posted by [mirek](#) on Wed, 22 Mar 2006 12:51:22 GMT

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

The purpose was not to replace .lay with XML files.

The purpose was to demonstrate that the basic U++ design idiom does not limit you to create your dialogs dynamically - this argument really appears quite often. In other, just because you CAN place widgets on that stack, you are not prohibibited to put then on the heap - or solve the problem by putting them to the Array.

Sure, I could implement all that you suggest, but that would make example ten times bigger, not something I would like to put inside reference example.

Perhaps I should have choosed different format from XML to make intentions of the example clear.

Mirek

Subject: Re: Interesting....

Posted by [mr_ped](#) on Wed, 12 Apr 2006 10:44:11 GMT

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

Mirek,

it's not about example, the one you posted is just fine IMHO.

It's about having some ready-to-use package to support XML dynamic menus right after U++ install.

I don't know enough about XML and layout to judge whether such package would save some development time. I thought from the basic premise about "widgets library" that the XML parser and the U++ library should be both powerful and flexible enough to make such package useless, because it's just few lines of code to put those two to cooperate and do what the developer want, right?

Than again from your comment about the example being lot bigger it looks like this is not the case.

Question is if you can design such package which would cut down development time for "XML dynamic menus" considerably, yet it will be flexible enough to cover all needs of such developer.

I'm definitely not the right person to even think about such design or just about possibility of such design.

Yet someone experienced with XML and UI controls of U++ (and generally UI programming experience) should probably look into this, if such interface can be designed and what work it can cut down without limiting the flexibility. (as the XML is flexible hell a lot ... only U++ UI controls are "set in stone" in this case, and that's true only as long as you don't support derived classes, which would be very likely "a must have" anyway.)

Subject: Re: Interesting....

Posted by [mirek](#) on Wed, 12 Apr 2006 11:20:48 GMT

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

Ah,

well, it is a little bit hard to explain, but....

U++ layouts are designed in a way that leads to "dialogs as structs". Means each layout naturally generates some C++ struct that contains widgets as members. This is one of dominant reasons why coding in U++ is so fast.

Now with dialogs stored in XML, this feature would be lost, because you would have to dynamically create dialogs by parsing XML and best you could hope for is to access widgets using some sort of textual IDs. U++ productivity gone.

While it is quite possible to implement this in U++, the question is why? If some real world app needs this, it will have most likely other very specific requirement, satisfying them would be most

likely harder than "XML dialog" skeleton code, which really is quite primitive.

Just a side note - I have couple of applications that store sort of dialog layouts in Oracle. Then others that store dialog layouts in proprietary text format. Still I do not see any advantage to provide some fixed facilities for dynamic dialogs.

Mirek

Subject: Re: Interesting....

Posted by [tojocky](#) on Tue, 27 Oct 2009 08:44:23 GMT

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

luzr wrote on Wed, 12 April 2006 14:20Ah,

well, it is a little bit hard to explain, but....

U++ layouts are designed in a way that leads to "dialogs as structs". Means each layout naturally generates some C++ struct that contains widgets as members. This is one of dominant reasons why coding in U++ is so fast.

Now with dialogs stored in XML, this feature would be lost, because you would have to dynamically create dialogs by parsing XML and best you could hope for is to access widgets using some sort of textual IDs. U++ productivity gone.

While it is quite possible to implement this in U++, the question is why? If some real world app needs this, it will have most likely other very specific requirement, satisfying them would be most likely harder than "XML dialog" skeleton code, which really is quite primitive.

Just a side note - I have couple of applications that store sort of dialog layouts in Oracle. Then others that store dialog layouts in proprietary text format. Still I do not see any advantage to provide some fixed facilities for dynamic dialogs.

Mirek

An interesting discussion i missed here.

About dynamic dialogs...

The main reason of dynamic dialogs is possibility to separate programming work by design work. With dynamic dialogs will be possibility to integrate styles too (visual styles)!

I think that is more important to make work easy even if is detrimental for running speed.

With this possibility will be possible to integrate forms in web browser! It is Not bad idea!

Ion Lupascu (Tojocky)

Subject: Re: Interesting....

Posted by [andrei_natanael](#) on Tue, 27 Oct 2009 09:59:53 GMT

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

tojocky wrote on Tue, 27 October 2009 10:44luzr wrote on Wed, 12 April 2006 14:20Ah,

well, it is a little bit hard to explain, but....

U++ layouts are designed in a way that leads to "dialogs as structs". Means each layout naturally generates some C++ struct that contains widgets as members. This is one of dominant reasons why coding in U++ is so fast.

Now with dialogs stored in XML, this feature would be lost, because you would have to dynamically create dialogs by parsing XML and best you could hope for is to access widgets using some sort of textual IDs. U++ productivity gone.

While it is quite possible to implement this in U++, the question is why? If some real world app needs this, it will have most likely other very specific requirement, satisfying them would be most likely harder than "XML dialog" skeleton code, which really is quite primitive.

Just a side note - I have couple of applications that store sort of dialog layouts in Oracle. Then others that store dialog layouts in proprietary text format. Still I do not see any advantage to provide some fixed facilities for dynamic dialogs.

Mirek

An interesting discussion i missed here.

About dynamic dialogs...

The main reason of dynamic dialogs is possibility to separate programming work by design work. With dynamic dialogs will be possibility to integrate styles too (visual styles)!

I think that is more important to make work easy even if is detrimental for running speed.

With this possibility will be possible to integrate forms in web browser! It is Not bad idea!

Ion Lupascu (Tojocky)

I don't see how you separate the code from the app design. You write handlers for widgets events. Even if you create a descriptive form of a window in XML file you still should parse that file, create widgets based on what you get from that file and your application should do something when user push a button or something. So the design of application imply also code.

How you will integrate styles? You cannot write CSS style attached to XML because U++ does know nothing about it.

I see the only benefit of a XML based design only when you write plug-ins which know nothing about your GUI. You write a XML file with description of (allowed) widgets in it(containing widget

name, position, etc) and then write event handlers(callbacks). Your app analyze the XML file, create a proper window based on info found in it and when an event occur from that window widgets it call your callbacks in plug-in, and your plug-in is doing his work. So, your plug-in may use the GUI without knowing nothing about it (you may use U++, Qt, GTK+, etc) and you don't have to change code in widget if for example you changed a widget name in your GUI library.

Subject: Re: Interesting....

Posted by [mdelfede](#) on Wed, 28 Oct 2009 12:34:57 GMT

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

I also don't see much point on having dialogs/controls in XML, or, weel, just a possible point, the dialog testing phase, on which you can "look" at dialogs without using them.

May be somehow useful, indeed, but it's also easy to write few upp code lines and show the same dialog using upp.

Another possible use would be to make dialogs with another app (which ???) and then export them to upp.... but then you'll have to connect dialogs to code anyways.

I'd see more useful some tool to convert dialogs from other toolkits to Upp. That would spare some porting time.

Ciao

Max
