Subject: How to program real U++ applications
Posted by koldo on Mon, 22 Mar 2010 14:55:58 GMT
View Forum Message <> Reply to Message

Hello all

I would like to ask you how you design professional U++ applications.

Thinking in a GUI application with a main window, how do you connect classes with main class containing main Serialize/Xmlize configuration:

- Main class is a global variable.
- You include in classes constructor as argument a pointer to main class
- You access from a subclass the main class using dynamic_cast<MainClass *>(GetParent()->GetParent()-> ... as many times as necessary.

Subject: Re: How to program real U++ applications
Posted by mdelfede on Mon, 22 Mar 2010 15:27:58 GMT
View Forum Message <> Reply to Message

I usually separate configuration from main class, so a "Settings" class wich contains all global settings. Example :

In "settings.h"

```
class Settings
{
   private:
......(all app wide settings here)

   public:
......(all getters/setters for public accessible settings)

};

Settings &globalSettings();
```

and, in "settings.cpp"

```
Settings &globalSettings()
{
   static Settings settings;
```

```
    return settings;
}
```

So the only public stuff is the global 'globalSettings()' function wich gives access to app-wide settings.
In settings constructor you can then put code to load (Xmlize) status, and in destructor code to save (Xmlize) status.

Ciao

Max

Subject: Re: How to program real U++ applications
Posted by mdelfede on Mon, 22 Mar 2010 15:33:26 GMT

View Forum Message <> Reply to Message

Another stuff is when you want to save app data, not just settings.
That depends too much upon your app structure....
In my app I've got files containing a 'fixed' part (customer name, site and so on...) and a 'variable' part wich is composed by an array of classes derived from a main (Page, in my app) base class, so a polymorphic class array.
For 'fixed' part I just xmlize single values, and for 'variable' part..... also Xmlize, but the full array using PolyXML bazaar stuff. That makes really easy to add new data to the application with no need to touch at a single line in main app class.

Ciao

Max

Subject: Re: How to program real U++ applications
Posted by mirek on Fri, 26 Mar 2010 13:32:36 GMT

View Forum Message <> Reply to Message

koldo wrote on Mon, 22 March 2010 10:55Hello all

I would like to ask you how you design professional U++ applications.

Thinking in a GUI application with a main window, how do you connect classes with main class containing main Serialize/Xmlize configuration:

- Main class is a global variable.

Never did that.

Quote:
- You include in classes constructor as argument a pointer to main class

Sometimes it is needed. Usually not.

Quote:
- You access from a subclass the main class using dynamic_cast<MainClass *>(GetParent()->GetParent()-> ... as many times as necessary.

Never did that.

Note there is nice global serialization system that perhaps is able to solve your issue.

Mirek

Subject: Re: How to program real U++ applications
Posted by zsolt on Fri, 26 Mar 2010 14:30:39 GMT

View Forum Message <> Reply to Message

koldo wrote on Mon, 22 March 2010 15:55Hello all

I would like to ask you how you design professional U++ applications.

Thinking in a GUI application with a main window, how do you connect classes with main class containing main Serialize/Xmlize configuration:

- Main class is a global variable.
- You include in classes constructor as argument a pointer to main class
- You access from a subclass the main class using dynamic_cast<MainClass *>(GetParent()->GetParent()-> ... as many times as necessary.

I usually use the global config scheme as well:

In your dialog cpp:

```
#define CONFIG_KEY "YourDialogClassName"

INITBLOCK {
 RegisterGlobalConfig(CONFIG_KEY);
}
```

before showing your dialog:

```
 LoadFromGlobal(*this, CONFIG_KEY);
```

After closing it:

```
 StoreToGlobal(*this, CONFIG_KEY);
```

And in your main window's Serialize method will be like this:

```
void MainWindow::Serialize(Stream& s)
{
 int version = 1;
 s / version;

 SerializePlacement(s);
 some_member.Serialize(s);
 SerializeGlobalConfigs(s);

 if(version<2) return;
 //this is used in newer versions
}
```

Subject: Re: How to program real U++ applications
Posted by koldo on Fri, 26 Mar 2010 19:01:45 GMT

View Forum Message <> Reply to Message

Thank you Massimo, Mirek and Zsolt

Do you know it this global serialization is available for Xmlize?

Subject: Re: How to program real U++ applications
Posted by mdelfede on Fri, 26 Mar 2010 20:35:05 GMT

View Forum Message <> Reply to Message

koldo wrote on Fri, 26 March 2010 20:01Thank you Massimo, Mirek and Zsolt

Do you know it this global serialization is available for Xmlize?

I don't think so... That's one of the reasons I use global Settings object.
BTW, I still use binary way/global serialization, but just for dialog placements/settings, I think app settings are better if centralized on a single object.

Ciao

Max

Subject: Re: How to program real U++ applications
Posted by koldo on Fri, 26 Mar 2010 22:07:35 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 26 March 2010 21:35koldo wrote on Fri, 26 March 2010 20:01Thank you Massimo, Mirek and Zsolt

Do you know it this global serialization is available for Xmlize?

I don't think so... That's one of the reasons I use global Settings object.
BTW, I still use binary way/global serialization, but just for dialog placements/settings, I think app settings are better if centralized on a single object.

Ciao

Max

Thank you Massimo

I think I will use your focus as I like Xmlize.

Thank you all again