Subject: Initial settings for U++ application
Posted by nlneilson on Sat, 10 Jul 2010 21:59:19 GMT
View Forum Message <> Reply to Message

How can the settings of a U++ app be be saved/implemented the next time the app is run?

I think it uses an .ini file in C/C++. In Java an .xml config file can be used.

It is an app ported from Java and works great in U++.

There may be ~20 options for settings ranging from meters or feet to the default directory for files.

Subject: Re: Initial settings for U++ application
Posted by dolik.rce on Sat, 10 Jul 2010 23:05:27 GMT
View Forum Message <> Reply to Message

Hi,

There is many options. All you have named and even more, though I would not recommend some of them (I really don't like apps which store their settings in windows registry) Anyway, you can choose whatever format you like. One more option is to use Serialize() methods to create binary configuration file.

One useful function you might want to know about (if you don't) is ConfigFile(). That will help you to determine the correct location.

For complex solution you can have a look at theide. Search for RegisterWorkspaceConfig, SerializeWorkspaceConfigs and sWorkspaceCfg.

In my opinion, for simple apps, with low and fixed number of options, the best solution is simple text file. Something like ini, or even simpler.

When you choose the format, we can help you in greater detail

Best regards,
Honza

Subject: Re: Initial settings for U++ application
Posted by nlneilson on Sun, 11 Jul 2010 00:22:19 GMT
View Forum Message <> Reply to Message

Thanks Honza

dolik.rce wrote on Sun, 11 July 2010 01:05(I really don't like apps which store their settings in windows registry

Serialize() methods

ConfigFile()

In my opinion, for simple apps, with low and fixed number of options, the best solution is simple text file. Something like ini, or even simpler.

I definitely do not want to tinker with the registry!

A simple text file would seem to be the easiest.

Here is a few that I have:
Set units m, km, ft, mi or nmi - - un = 0 to 4
Set deg, dm or dms - - dms = 0 to 2
Set precision - - dp = 3, 4, 5, 6 or 8
Set goto direct/flyto - - gt = 0, 1
Set draw - - dr = 0, 1
fn <<= Nvl(r_d, recent_dir);

There are several others that just have int variables as above.

So all the config file needs is:

```
un = 2
dms = 1
dp = 4
gt = 0
dr = 1
...
...
recent_dir = c:\wwdata
```

So just a text file that would be read and set the variables.
On exit write the variable values to the text file.

Neil

Subject: Re: Initial settings for U++ application
Posted by dolik.rce on Sun, 11 Jul 2010 08:40:09 GMT
View Forum Message <> Reply to Message

Hi Neil,

Some time ago I write a simple class for similar purpose. I actually used it even for other things, like passing large number of parameters to a function.

It saves key-value pairs as strings. So you have to make all the conversions yourself, but if you have low number of options to save it should not be a problem.

The interface is simple, you will probably need only few functions:
config cfg(filename); //constructor which loads a file
cfg.Load(filename); //load a file into existing config object
cfg.Get(key,default); //get the value from underlaying file
cfg(key,default); //same as Get(), shortcut
cfg.Set(key,value); //store value to config object
cfg.Save(); //write the stored values to the file
It has even some more capabilities, like saving arrays and tables (2D arrays), but based on your description you won't need that. The resulting file is human readable and supports comments (#).

I never had time to really polish it, so it might not follow the U++ coding style. Also, there might be some bugs, since I just quickly stripped out some of my app specific stuff. But don't be afraid, even though this disclaimer, it should do the job

Small example of usage is enclosed with the class.

Bye,
Honza

File Attachments

1) config.zip, downloaded 292 times

Subject: Re: Initial settings for U++ application
Posted by nlneilson on Mon, 12 Jul 2010 01:35:27 GMT

View Forum Message <> Reply to Message

Honza, I will try your example.

Something like this should work.

Neil

Subject: Re: Initial settings for U++ application
Posted by nlneilson on Thu, 22 Jul 2010 05:58:21 GMT

View Forum Message <> Reply to Message

I tried for a few hours to get the sample to run with no luck.
Linking errors, probably a simple fix for someone that knows what they are doing.

I ended up creating a text config file that is read when the app starts.
Each line sets a variable.
Clicking the red x at the top of the window exits the app without changing the config file.
Clicking "File" -> "Save Exit" saves any changes to the config file.

Works OK.

I did run into one glitch.
I use atof for other parts of the app that works fine.
atoi for the integers gave errors.
Had to compare characters.

```
 while(!in.IsEof()){
  String Ln = in.GetLine();
  if (i==0) {
      if (Ln=="0") Setm();
      if (Ln=="1") Setkm();
      if (Ln=="2") Setft();
      if (Ln=="3") Setmi();
      if (Ln=="4") Setnmi();
  }
  if (i==1) {
                    ....
```

Subject: Re: Initial settings for U++ application
Posted by koldo on Thu, 22 Jul 2010 06:27:00 GMT
View Forum Message <> Reply to Message

Hello nlneilson and Honza

To store the settings of an U++ application I love Xmlize functions (sample here http://www.ultimatepp.org/reference$Xmlize$en-us.html).

It is xml so it is text based. However it can be easily mixed with other functions to be encrypted.

Subject: Re: Initial settings for U++ application
Posted by andrei_natanael on Thu, 22 Jul 2010 20:53:11 GMT
View Forum Message <> Reply to Message

If it's not necessary to use a text file(editable by user) why not use Serialize() with LoadFromFile/StoreToFile?

Subject: Re: Initial settings for U++ application
Posted by nlneilson on Sat, 31 Jul 2010 14:49:51 GMT
View Forum Message <> Reply to Message

The text file does not need to be editable by the user.
I need to look at the code to understand what each line represents, I do not have it commented. It was something I was able to figure out and get to work.

"Serialize() with LoadFromFile/StoreToFile" may work also, I will look at an example that uses it.