Subject: File system library refactoring
Posted by cbpporter on Mon, 19 Nov 2007 21:11:51 GMT
View Forum Message <> Reply to Message

While U++ has a good file system interaction library, I think that that the non-visual parts of it are pretty low level and the visual ones also could be improved a lot.

So I started working on some high level path manipulation classes. The class PathInfo has methods such which will initialize with some OS aware abstract path:

```
PathInfo& FromDir(const WString &path);
PathInfo& FromFile(const WString &path);

PathInfo& FromAppDir(const WString &path);
PathInfo& FromUserDir(const WString &path);
PathInfo& FromTempDir(const WString &path);
```

It will also feature manipulation methods such as:

```
PathInfo& AddPath(const WString &path);
PathInfo& SetPath(const WString &path);
PathInfo& AddDir(const WString &path);
PathInfo& SetDir(const WString &path);
```

PathInfo will maintain a state and will not allow operations which are invalid on the given OS. For example, using AddDir on an object which has not been initialized with one of the From*** methods will leave the object in an invalid state and in debug mode will fail an assert.

This way, paths can be constructed in a more elegant and intuitive way that using the current functions which operate on Strings. For example, you could determine the path to store or load you ini file this way:

```
PathInfo iniFile;
iniFile.FromUserDir().AddPath("config").AddFile("config.ini");
```

And since the only way to get a valid PathInfo object is from a function or control that returns such, these objects will be safe and the code cross platform. And speaking of controls, I plan to extend/create another FileList, which will allow to implement a FileSel dialog only with a few lines of code, but can be also used as a tree or a folder select component. I would also like to emulate the native OS's look whenever possible. This means creating some dummy folders like "Computer" under Vista and inserting the real items into these. These will be purely cosmetic changes.

But for now I need to finish PathInfo first. What do you think about it? And also, I used WString instead of String.

Subject: Re: File system library refactoring
Posted by cbpporter on Thu, 22 Nov 2007 14:08:29 GMT
View Forum Message <> Reply to Message

I have a little problem. How can I use WinAPI functions which are not defined already. Also, I need the function SHGetFolderPath which uses different dll under different windows versions. And also, how can I determine the user's home directory under Linux.

And another question. I'm not that comfortable yet with pick, so If my structure has a Vector and I need the object to be copyable, must I use optional deep copy?

Subject: Re: File system library refactoring
Posted by mrjt on Thu, 22 Nov 2007 15:46:37 GMT
View Forum Message <> Reply to Message

Not sure if this is any use, but inside the zip are some functions I wrote for getting Windows systems folder paths. I never really finished what I was using them for but I think the code works correctly, though it reads the path from the registry instead of using SHGetFolderPath path. I did some research and had a good reason for doing this, but I can't remember what it was .

The documentation doesn't mention SHGetFolderPath needing a different library on different versions, are you sure? And to add an extra windows lib you should just have to add it the package organiser.

And the home directory on Linux can be found using GetHomeDirectory().

James

File Attachments

1) Dirs.zip, downloaded 638 times

Subject: Re: File system library refactoring
Posted by cbpporter on Fri, 23 Nov 2007 11:20:33 GMT
View Forum Message <> Reply to Message

Thank you for the files! I guess using registry is just about as OS dependent as SHGetFolderPath function, so it could be okay. For now I'll finish the basic framework and add these functions later.

AFAIK, SHGetFolderPath in Windows versions <= 4 is in SHFolder.dll and in >=5 in Shell32.dll. I don't know if there must be a special configration step if you just link with shell32.lib. I'll have to try.

I'm also posting a preview the sources to show what I'm trying to achieve together with some nonsensical example:

PathInfo p;
p.FromFile("c:\\miu\\matz\\miau\\pisu.txt").AddDir("\\a/").AddDir("b\\c\\d").SetFile("pisu.exe").SetDir(4, "AA");

will result in:
c:\miu\matz\miau\AA\b\c\d\pisu.exe

PS: The rar contains some extra files from my future CtrlEx, just ignore them .

File Attachments
1) CtrlEx.rar, downloaded 340 times

Subject: Re: File system library refactoring
Posted by mrjt on Fri, 23 Nov 2007 11:34:33 GMT
View Forum Message <> Reply to Message

I'm not sure Upp actually supports versions that old (Windows 4.0 = Windows 95 I think). CtrlLib certainly doesn't, but CtrlLib might.

I personally wouldn't worry about it unless you know you are going to use it on anything that old.