
Subject: File Chooser closing

Posted by [nneilson](#) on Wed, 05 Jan 2011 22:50:38 GMT

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

I searched but could not find an answer, I may just be missing something.

When the file choosing window is opened and without picking a file when the "Cancel" or red "X" is clicked to close the file chooser it opens a file. How can this window be closed without opening a file?

```
fn <=<= Nvl(r_d, recent_dir);
if(fn.ExecuteOpen()){ // file chooser
    Data<=<=Value(fn);
    r_d = fn;
    recent_dir = ~fn;
}
```

Neil

Subject: Re: File Chooser closing

Posted by [dolik.rce](#) on Thu, 06 Jan 2011 11:01:28 GMT

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

Hi Neil,

I failed to reproduce the problem you describe. The way you use ExecuteOpen() seems correct, it should return false when closing the window or hitting Cancel. I tested with GCC and with MinGW. Are you using FileSel or FileSelector? Maybe if you post a complete, compilable testcase that shows the wrong behavior we can figure out what is the problem...

Also, I don't really understand what are you trying to do with the r_d and recent_dir. Both fn and ~fn return the same thing, the difference being that once it is string, and the second is value, but in the end they are used in Nvl, so I deduce they are same type or they are converted to the same type upon use, so I don't see much sense to have them both. But that is not really related to the problem.

Best regards,
Honza

Subject: Re: File Chooser closing

Posted by [nneilson](#) on Thu, 06 Jan 2011 14:52:09 GMT

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

Attached is a .zip that has the full package for an app that has two instances of a file chooser.

FileSel or FileSelector?, Nvl?

I found an example that worked to get a file and have been using the same code since then. What it does or how it does it I don't know, I could not find a help file that explained it.

I am using MSC9.

Neil

edit: removed .zip, bad example

Subject: Re: File Chooser closing

Posted by [dolik.rce](#) on Thu, 06 Jan 2011 16:39:57 GMT

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

I still can't reproduce what you describe... Unfortunately I don't have MSC to test (actually I don't even have windows). Maybe someone else out there could give it a try with MSC?

nneilson wrote on Thu, 06 January 2011 15:52FileSel or FileSelector?, Nvl?

FileSelector uses windows API to open native windows file selector.

FileSel is pure U++ file selector dialog. It is typedefed to FileSelector on non-windows systems.

You can use FileSel on windows, but to be more user friendly I would choose FileSelector...

Nvl is a utility functions that checks if the first parameter is not Null. If it is, it returns the second argument, otherwise it returns the first one.

Anyway, for your needs, it is just fine to call something like

```
void OpenBaseFile() {  
    fn1.ActiveDir(recent_dir);  
    if(fn1.ExecuteOpen()){ // file chooser  
        filename1 = fn1;  
        BaseFile<=<Value(fn1);  
        recent_dir = fn1.GetActiveDir();  
    }  
}
```

In the attached file you will find even more elegant solution. I also fixed a couple of glitches in your code. All the changes are marked using a comment with my nick and explained.

Honza

File Attachments

1) [DGPSmerge.cpp](#), downloaded 627 times

Subject: Re: File Chooser closing
Posted by [nneilson](#) on Thu, 06 Jan 2011 19:01:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Honza

I tried it with the changes and it works OK and clicking Cancel or the red X exits the file chooser the way it should.

Looking at my code it is a bit of a mess, I had renamed another app and made changes from there. Probably 90% of the variables are not used in this app.

There are a few things noticed so far:

The file chooser window is partially off of the upper left of the screen, hard to move it.

The .cfg file is saved in the directory of the files opened, it should be in the same directory as the .exe.

All minor and can be easily changed.

The EditFields for the files opened are editable for the only purpose to copy, no data will be input there manually but the ability to input the file name will be handy in another app.

The only data that will be manually input is the base file location, extra error handling which was done in another app can be added.

I have not tried this in Linux yet with wine, if there is a glitch there using FileSelector then FileSel will work fine.

I noticed several other changes that will be improvements.

It will take me some time to understand all the changes.

The problem I had exiting the file chooser is solved.

Neil

Subject: Re: File Chooser closing
Posted by [nneilson](#) on Thu, 06 Jan 2011 23:42:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Thu, 06 January 2011 17:39I still can't reproduce what you describe...

That was not a good example as the problem only happened about half the time.

Here is an example that has the problem every time after the first time a file is opened.

File->Save/Exit will save the recent_dir to a .cfg file.

Also included in the .zip is ReplayTest.txt. This file can be opened and all this example does is read the first ten lines with sleep in between and displays that line in the top EditField. I had to include the thread so the display is visible when it is running, other than that it is as simple as I

could make it and still show the problem.

I have not made the changes from your previous example yet but will.

THISBACK1 rather than THISBACK ??

(Ctrl* ctrl) ??

*ctrl<=<fn0.Get(); ??

They work but I am not familiar with these.

Neil

File Attachments

1) [Replay.zip](#), downloaded 421 times

Subject: Re: File Chooser closing

Posted by [nneilson](#) on Fri, 07 Jan 2011 00:54:20 GMT

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

I made some changes to the .cpp but apparently not enough.

The problem is still present.

File Attachments

1) [Replay.cpp](#), downloaded 426 times

Subject: Re: File Chooser closing

Posted by [dolik.rce](#) on Fri, 07 Jan 2011 10:20:55 GMT

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

Hi Neil,

The only problem I see in your code is that the work thread is started every time you call OpenFile(), while it should be called only when ExecuteOpen() returns true: void OpenFile(){

```
fn0.ActiveDir(recent_dir);
if(fn0.ExecuteOpen()){
    FileEF<=<fn0.Get();
    recent_dir=fn0.GetActiveDir();
    work.Run(THISBACK(Work)); // <- moved here
}
//work.Run(THISBACK(Work));
```

}That might "look like" the file is opened even when you cancel the selection, because the replay happens unconditionally every time. Is that the problem you have? But that would not be problem

in the other application you provided as example...

If there is something else, than I'm seriously puzzled. It might be MSC related, in which case, I have no chance to figure it out.

Honza

Subject: Re: File Chooser closing
Posted by [nlneilson](#) on Fri, 07 Jan 2011 11:31:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Honza

That was the problem.

dolik.rce wrote on Fri, 07 January 2011 11:20 That might "look like" the file is opened even when you cancel the selection, because the replay happens unconditionally every time. Is that the problem you have? But that would not be problem in the other application you provided as example...

"...because the replay happens unconditionally every time."

I was calling the work thread outside of the:
`if(fn0.ExecuteOpen()){`
`...`
`}`

The previous example did not use a thread. Even with 8 hours of data it only took a few seconds to merge.

There is a separate call to merge by clicking File->Merge so it was not automatic.

`bar.Add("Merge Files", THISBACK(Merge));`

If the file chooser was closed with "Cancel" without picking a file the variables for the out, base and roam files were still set even if the data in the EditField was deleted.

When the app was started the .cfg file set the (recent_dir) to when the app was previously run, opening the file chooser and closing with "Cancel" set whatever file was to be picked to that of the previous run, (recent_dir), and that was only when a file was not picked. Normally this would not be much of a problem but testing found this error.

I am used to something like `if(fn0)` and `if(!fn0)`
The `if(fn0.ExecuteOpen())` added to my confusion.

Calling the work thread from inside the `if()` block fixed things.
I have two other apps with the same fix.

Much Thanks!

Neil

