Subject: TabBar. Uninitialized member
Posted by Novo on Thu, 29 Jan 2009 04:39:47 GMT
View Forum Message <> Reply to Message

In the code below, which is called indirectly from TabBar::TabBar(), sz is not initialized.

```
bool TabScrollBar::IsScrollable() const
{
 // Note: sz already 'fixed'
 return total > sz.cx && sz.cx > 0;
}
```

That can be reproduced using DockingExample1. As long as Size() is not initialized by design, it should be initialized explicitly in TabBar::TabBar().

Subject: Re: TabBar. Uninitialized member
Posted by Mindtraveller on Wed, 08 Apr 2009 23:41:00 GMT
View Forum Message <> Reply to Message

One more little bug. Happens when all the tabs are closed and after that parent window is being closed. Parent window invokes Accept() for all the children, and TabBar::GetData() is called. Due to one can`t set TabBar::SetCursor(-1) before all it`s tabs are closed, TabBar tries to Get(0) and fails with ASSERT.
There are many ways to correct this. The simplest one is checking if there are any tabs in TabBar before trying to Get anything:

TabBar.h, line 268:

```
 virtual Value GetData() const   { return HasCursor() ? Get(active) : Value(); }
```

change with:

```
 virtual Value GetData() const   { return (GetCount() && HasCursor()) ? Get(active) : Value(); }
```

Subject: Re: TabBar. Uninitialized member
Posted by Mindtraveller on Fri, 10 Apr 2009 16:19:07 GMT
View Forum Message <> Reply to Message

A little proposal.
Each tab corresponds to some object. Adding a tab is made with adding some Value to TabBar internal list. Most of time this Value is String, which is drawn inside tab.

So imagine that user`s made something with one of controls associated with current tab. What do I do?
1) First of all, I pass my internal objects Vector to determine which of them corresponds to current tab
2) Do some action

I think it will be better to add tabs with a pair of (Value,Value) (like in DropList). One may hold "title" information (like String) while another will hold some reference info (like index, pointer or event One<MyTabbedClass>).

Subject: Re: TabBar. Uninitialized member
Posted by koldo on Wed, 20 May 2009 13:26:07 GMT
View Forum Message <> Reply to Message

Hello all

I fully agree with Mindtraveller

Mindtraveller wrote on Fri, 10 April 2009 18:19A little proposal.
Each tab corresponds to some object. Adding a tab is made with adding some Value to TabBar internal list. Most of time this Value is String, which is drawn inside tab.
So imagine that user`s made something with one of controls associated with current tab. What do I do?
1) First of all, I pass my internal objects Vector to determine which of them corresponds to current tab
2) Do some action

I think it will be better to add tabs with a pair of (Value,Value) (like in DropList). One may hold "title" information (like String) while another will hold some reference info (like index, pointer or event One<MyTabbedClass>).

I am using TabBar for opening files. To manage them I need full path but I would like to put in tab String only the file name.

Now as far as I know this is not possible so I have to put the full path in the tab String.

Best regards
Koldo

Subject: Re: TabBar. Uninitialized member
Posted by mrjt on Wed, 20 May 2009 14:56:36 GMT
View Forum Message <> Reply to Message

Quote:I am using TabBar for opening files. To manage them I need full path but I would like to put in tab String only the file name.

Now as far as I know this is not possible so I have to put the full path in the tab String.

Actually, in the current SVN version what you want to do is pretty easy. You have three choices:
- Use a Display (will only work properly for a top-aligned TabBar, but that is almost all cases)
- Inherit and overload ParseLabel
- Inherit and overload PaintTabData (this is intended for more complex scenarios requiring left/right/bottom alignment support).
I would suggets using a Display, as I would personally like to get rid of ParseLabel. The Bazaar/QuickTabsTest/FileTabs class is an example of using the TabBar to implement TheIde QuickTabs. It also uses tab stacking and used the third method of Paint overloading to display additional stacking info.

Having said all of that, I actually agree with you. Up until now the package has largely been the result of just adding what myself and Unodgs needed on top of his original QuickTabs class without discussion since we didn't know anyone else was using it. Perhaps we should have that discussion now so we can tidy it up some? I'm happy to do some work on it next week (off to the Monacco GP in the meantime), and maybe even write some docs. But I'd like Uno's input before I start since I know he's using the TabBar in an Ide branch at the moment. I'll PM him.

Any other feature requests while we're at it?

Edit: And apologies to Mindtraveller for missing his posts before, the bug has been noted and fixed

---

Subject: Re: TabBar. Uninitialized member
Posted by koldo on Wed, 20 May 2009 15:39:20 GMT
View Forum Message <> Reply to Message

Hello mrjt

Thank you for your answer!. I would not like to press you but I like this control .

Well, I have done a new Display. It is not very clean as I do not want to change the Display. I want only to display other text.

One thing: I have done this fix:

TabBar& SetDisplay(const Display &d) { display = &d; Refresh(); return *this; }

The return *this; was not included in the original code.

Waiting for your code...

Best regards
Koldo

Subject: Re: TabBar. Uninitialized member
Posted by koldo on Wed, 20 May 2009 20:19:54 GMT
View Forum Message <> Reply to Message

Hello mrjt

Finally I have failed using a Display.

More things to add to the Santa letter...:

- An interesting feature would be to add to the "WhenClose" callback the possibility of avoiding the closing of the tab. Now also the tab is visually closed before WhenClose function is called.

- I would ask you to put an example of "grouping" and "stacking" features, as I do not understand them well.

Best regards
Koldo