

---

Subject: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Sun, 10 Apr 2011 02:57:57 GMT

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

---

In windows or certain language on Win32, before a window/container is closed, it query whether the window can be closed thus give a chance for the program to determine whether there are unsaved work, and check if the end user wants to save his/her work, discard it, or even cancel the close action (might be by accident).

What's the equivalence in U++?

I did some test in U++. I open some files (some are the essential libary files), and make some changes and click the little square on the tab. It silently decides to save the possibly unintentional changes and closes the current tab. I am afraid this is too bold; the traditional Windows' way, user consent before action, is more reasonable IMHO.

Thanks,  
Lance

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Sun, 10 Apr 2011 03:26:11 GMT

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

---

Then I find this code snippet in UWord

```
UWord::UWord()
{
.....
WhenClose = THISBACK(Destroy);
.....
}
void UWord::Destroy()
{
if(editor.IsModified()) {
switch(PromptYesNoCancel("Do you want to save the changes to the document?")) {
case 1:
Save();
break;
case -1:
return;
}
}
delete this;
}
```

WhenClose is a TopWindow callback, so I know how to guard it when the main window is to be closed. Now, like TheIDE, many files(slave ctrls) are in TabCtrl pages, what events should I hook to when a tab page is to be closed?

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Sun, 11 Dec 2011 17:28:28 GMT

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

---

OK. (one of?) The right way is to override the Accept virtual method of Ctrl-derivated class.

In my case, I use TabCtrl to host different pages that are derivated from ParentCtrl, in the overrided Accept, do something like:

```
bool MyPage::Accept()
{
    bool ok=true;

    if( DataIsChanged() )
    {
        switch(PromptYesNoCancel("Data Changed, do you want to save?"))
        {
            case -1: // Cancel;
                ok=false;
                break;
            case 1: // Yes
                if(!SuccessfullySaveD())
                    ok=false;
        }
    }
    return ok;
}
```

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Sun, 11 Dec 2011 17:37:11 GMT

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

---

Sorry, misinformation. Above only works when the application is closed. It failed to detect when the hosting tab page is closed.

---

---

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits  
Posted by [Wolfgang](#) on Sun, 11 Dec 2011 17:48:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Do you want to let your app ask if he want to save his changes then your tab changes?

---

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits  
Posted by [Lance](#) on Sun, 11 Dec 2011 22:57:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Thank you, Wolfgang!

If that's the only way. I would prefer not though. For example, I may be editing something in one tab page and occasionally switch to another tab page to consult some other information. Say I am doing an order, in the process I may become interested in a particular inventory item, e.g. its flow and current count, etc, I will double click the item to bring out a new tab page displaying the inventory activities information. It may not hurt that much to save partial edits but it's preferable not to commit until I (the end user) ask to, e.g. I may subsequently decide to cancel changes made in this session, it will be necessary to remember the edit/undo history if partial changes and saved in the editing process.

---

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits  
Posted by [Wolfgang](#) on Sun, 11 Dec 2011 23:14:30 GMT  
[View Forum Message](#) <> [Reply to Message](#)

I dont see your problem... if you change the active tab, you don't have to destroy the old one. This way you just have to show / hide the tab with the user changed information. And if you want to save them, you can use the destroy event of the mainwindows to check if something have changed - if yes, ask the user if you should save the changes.

---

---

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits  
Posted by [Wolfgang](#) on Mon, 12 Dec 2011 08:38:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

I dont know if I understand you the right way but I uploaded an example for you how I would handle the "saving" problematic..

A simple example project with 3 tabs, one for settings which is monitored for changes!  
Hope it helps!

## File Attachments

1) [lance.zip](#), downloaded 371 times

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Sender Ghost](#) on Mon, 12 Dec 2011 12:14:21 GMT

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

---

Hello, Lance.

Lance wrote on Sun, 10 April 2011 04:57

I did some test in U++. I open some files (some are the essential libary files), and make some changes and click the little square on the tab. It silently decides to save the possibly unintentional changes and closes the current tab. I am afraid this is too bold; the traditional Windows' way, user consent before action, is more reasonable IMHO..

Lance wrote on Sun, 10 April 2011 05:26

WhenClose is a TopWindow callback, so I know how to guard it when the main window is to be closed. Now, like TheIDE, many files(slave ctrls) are in TabCtrl pages, what events should I hook to when a tab page is to be closed?

In my opinion, TheIDE behaviour with TabBar is normal if you use some kind of source code management system (svn, git, etc.) to be able revert changes when needed.

Lance wrote on Sun, 11 December 2011 18:28

In my case, I use TabCtrl to host different pages that are derivated from ParentCtrl

Lance wrote on Sun, 11 December 2011 23:57

For example, I may be editing something in one tab page and occasionally switch to another tab page to consult some other information. Say I am doing an order, in the process I may become interested in a particular inventory item, e.g. its flow and current count, etc, I will double click the item to bring out a new tab page displaying the inventory activities information. It may not hurt that much to save partial edits but it's preferable not to commit until I (the end user) ask to, e.g. I may subsequently decide to cancel changes made in this session, it will be necessary to remember the edit/undo history if partial changes and saved in the editing process.

In the attachment you could find another example how to save/cancel, revert to default changes of TabCtrl, while using Serialize and storage data structure(s).

Also, take a look for CtrlRetriever example.

## File Attachments

---

1) [TabCtrlSerialize.zip](#), downloaded 348 times

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Mon, 12 Dec 2011 15:52:56 GMT

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

---

Hi Wolfgang & Sender Ghost:

Thanks for the help! I will try your codes in a while.

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Tue, 13 Dec 2011 00:15:11 GMT

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

---

I am sorry. The post was from long ago, I actually changed TabCtrl to TabBarCtrl to allow closing individual TabPages. Your samples are very interesting; the serialization part in particular is something I knew was useful but never have a chance to study.

Anyways, I figured out the thing I was looking for should be CancelClose gate of TabBar (parent of TabBarCtrl), hook up the following code fixed my problem, quite decently IMHO

```
bool MyMain::TabCancelClose(Value key)
{
    return !myTabBarCtrlInstance.FindCtrl(key)->Accept();
}
```

Each Complex Ctrl that's to be hosted in a Tab page can individually decide its Accept() logic. For example, my example code above will query user to see if he/she wants to save/discard/cancel when there are modifications, and takes action accordingly.

My only remaining question was, for the TabBar libary developers, should the above logic be built in, as it should the very purpose of Accept() in Ctrl's designers intention? Sorry I don't know enough to judge, just raising a question.

TabBar has CancelCloseSome and similar gates, I am not sure if they are of any interests. For my purpose, the above is exactly what I wanted.

Thanks again to Wolfgang and Sender Ghost.

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Tue, 13 Dec 2011 03:29:18 GMT

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

---

Above solution still has a flaw.

If the Ctrl (the outmost container) in a page is modified and the containing page is crossed (to initiate a close request), the user will be prompted whether he/she wants to save, discard edit or cancel the close action when overridden Accept() is called from within CancelClose gate. If user selects "no" to discard change (close without saving) and no action was taken to unset the modified flag, the Accept() gets called again before the Ctrl is destructed. A simple fix would be to always reset modified flag when the user chooses to discard edit. This way when Accept() is called at the moment of destruction, it always returns true so that it is silently disposed of as wished.

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [mdefede](#) on Fri, 16 Dec 2011 20:09:00 GMT

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

---

If I have did understand your problem, there's TabBarCtrl in Bazaar to which I added some time ago a couple of callbacks  
named CancelCloseSome and WhenCloseSome just for that purpose.

First is called BEFORE a tab or also many tabs are being closed, giving the ability to prompt for a file/list of files to be saved; the latter happens AFTER the tabs are closed and allows the application to finish closing / do some cleanup.

It was refactored by original maintainer, but those callbacks are still there, IIRC.  
I'm using a modified version of this control in my app and works quite good.

Max

Edit : I've read now one of your posts about CancelCloseSome and WhenCloseSome.... the purpose is the one I've explained above; you can close multiple tabs at once and those allows you to group the closing prompts.

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Sat, 17 Dec 2011 03:31:47 GMT

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

---

Hi Max:

Thanks for the reply!

Are you the current maintainer of TabBarCtrl? If yes, I would like to request a feature. Basically, let's say EditInt1 on Tab Page 1 is currently focused, then I switch to Tab Page2, I would expect some control in Tab Page2 to receive the key board focus. But EditInt1 will continue hold the key board focus until I, say, click EditString2 on Tab Page2. Then I switch back to Tab Page1, and of course, EditString2 will not release key board focus to EditInt1 as would be desirable in most cases.

The current implementation requires library users to write callbacks to remember and restore keyboard focus upon Tab changes. Not that it's hard to do, but it should not be necessary.

It's not only a problem with TabBarCtrl. In general, a container Ctrl should remember its focused child and restore focus to it when it receives focus. In the current implementation, if a window receives focus because of another window who had the focus was closed, its first focusable child will receive the focus instead of the previously focused child in that window. Having container Ctrl to remember and restore its focused child is too often requested by library users to leave it out.

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [mdelfede](#) on Sat, 17 Dec 2011 09:54:13 GMT

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

---

No, sorry... I'm not the maintainer of TabBarCtrl.

I'm using a custom version since long.

Max

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [Lance](#) on Sun, 18 Dec 2011 15:28:09 GMT

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

---

Interesting. To maintain a customized version, what do you need to do when there are updates for upp? Simply overwrite with your own version or do you need to check line by line every time? Or is there a tool for that purpose?

---

---

Subject: Re: CanClose? Mechanism to prompt user to save edits

Posted by [mdelfede](#) on Sun, 18 Dec 2011 15:35:16 GMT

I do nothing because my customized version is all what I need

Normally I avoid this, but the original maintainer made some changes that were incompatible with my app so, having no time to modify my code, I had to gather my previous mods and stay with it.

If you have to start from scratch, I suggest you to use bazaar code. Mine is not polished and has no great advantages.

About keeping in sync with upp.... there are no special tools, we're using svn. Usually each piece of bazaar has its own maintainer (not everywhere, but usually....) so when we need some mods we ask him if it's ok to add OR we do ourselves, but taking care that they'll not break anything.

The case with TabBarCtrl was peculiar because I needed it, it was not there, so I added to original code, but the "owner" took it in charge later on and rewrote it to its taste. The only things that were kept are the CanCloseSome() and WhenCloseSome() callbacks I added.

BTW, the mod you're requesting for makes sense and should be almost trivial.... I don't know if original maintainer is still working on it, but you can ask here.

Max

---