Subject: menu Tree
Posted by nlneilson on Mon, 25 Apr 2011 05:26:38 GMT

View Forum Message <> Reply to Message

Hope this will help someone.

I forgot how it was done so spent a couple hours searching.
Nothing was found in the /examples or /tutorial but in /reference/TreeCtrl that made my head hurt,
too much stuff to hide what I was looking for, a simple example.
```
 void MainMenu(Bar& bar) {
  bar.Add("File", THISBACK(FileMenu));
  bar.Add("Layers", THISBACK(LayersMenu));// this is a sub menu

  ...
 // Layers sub-menu
 void LayersMenu(Bar& bar) {
  bar.Add("FAA", THISBACK(FAAMenu));// this is a sub sub menu
  bar.Add("Marble", THISBACK(SetMarble)).Check(marbleX);

  ...

 // FAA sub-menu
 void FAAMenu(Bar& bar) {
  bar.Add("WAC", THISBACK(Setwac)).Check(wacX);
  bar.Add("WAC sep", THISBACK(WACsepMenu));// this is a sub sub sub menu

  ...
```

After that the code was easy to implement.

Neil


```
File Attachments
```
1) EFB_menu.jpg, downloaded 426 times

Subject: Re: menu Tree
Posted by nlneilson on Wed, 11 May 2011 23:45:09 GMT

View Forum Message <> Reply to Message

Is there a way to keep the layer tree visible as long as the mouse is over it.

A user could click to enable 1L then 1R, 2L and 2R without chasing down the layer tree again.

This is for the U.S. FAA Charts. The merged set is one click but for the charts with the full text
borders they are set up in a checkerboard layout, hence the 1L, 1R, 2L and 2R.

Upp has been working great for my apps.

Neil

Subject: Re: menu Tree
Posted by nlneilson on Thu, 12 May 2011 18:04:28 GMT
View Forum Message <> Reply to Message

After debugging through the code keeping the Menu Tree in scope could be a real bag of worms at least for me.

Another way would have an informative text box similar to what was done here except without the THISBACK to or on the text.

Is there a way to do this in Upp?

Neil

Subject: Re: menu Tree
Posted by nlneilson on Sat, 14 May 2011 02:17:25 GMT
View Forum Message <> Reply to Message

```
    bar.Separator();
```

This seems to function as needed to add a line without THISBACK being involved.

Now if the Separator line could be replaced with text that would work.

Any suggestions?

Neil

Subject: Re: menu Tree
Posted by nlneilson on Sat, 14 May 2011 08:41:12 GMT
View Forum Message <> Reply to Message

I can't figure out how it can be done in Upp.

In Java

// This makes the menu item on a list

```
  clearMenuItem = new JRadioButtonMenuItem("Clear");
  clearCtrl.add(clearMenuItem);
```

// This concerns the action taken when clicked.
// If these lines are not included then there is just a line of text ("Clear")

```
  clearMenuItem.addActionListener(new java.awt.event.ActionListener() {
   public void actionPerformed(java.awt.event.ActionEvent evt) {
      try {
       ConnSock.sock("clear96");
    } catch (IOException e1) {}
   }
  });
```
This is what I am trying to do in Upp.
Just informative lines of text with no action, THISBACK, or Callback.

Maybe overloading bar.Add() to accept just one argument, the text, without the THISBACK.

Subject: Re: menu Tree
Posted by Sender Ghost on Sat, 14 May 2011 12:57:28 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Sat, 14 May 2011 10:41
Just informative lines of text with no action, THISBACK, or Callback.

Hello, Neil.

You can add Ctrl(s) to Bar with using Add(Ctrl& ctrl) or AddNC(Ctrl& ctrl) functions. For "informative" text, you can use Label Ctrl, for example:

#include <CtrlLib/CtrlLib.h>

using namespace Upp;

```
class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();

 MenuBar menu;
 Label label;
```

```
 void MainBar(Bar& bar);
 void FileBar(Bar& bar);
};

App::App()
{
 Title("MenuBar test application");
 CenterScreen().Sizeable().MinimizeBox().MaximizeBox();
 SetRect(Size(320, 240));
 AddFrame(menu);
 AddFrame(TopSeparatorFrame());
 menu.Set(THISBACK(MainBar));

 label.SetLabel("Some informative text");
}

void App::MainBar(Bar& bar)
{
 bar.Add("File", THISBACK(FileBar));
}

void App::FileBar(Bar& bar)
{
 bar.AddNC(label);
 bar.Separator();
 bar.Add("Exit", THISBACK(Close)).Key(K_CTRL_Q);
}

GUI_APP_MAIN
{
 App app;
 app.Run();
}
```

Subject: Re: menu Tree
Posted by nlneilson on Sat, 14 May 2011 16:10:29 GMT
View Forum Message <> Reply to Message

Thanks Sender Ghost, I will try
 label.SetLabel("Some informative text");

I got something to work in the mean time but the SetLabel may be better and less code.

Rather than clicking on 1L to turn on the layer that includes all in the text, which I wanted, one of

the names in the text needs to be clicked.

Neil

Subject: Re: menu Tree
Posted by nlneilson on Sun, 15 May 2011 06:01:05 GMT

Hi Sender Ghost

I tried what you suggested and thanks for posting the code.
It's much easier to do this:
  label_1L.SetLabel(\
"Atlanta\n\
Brownsville\n\
Dallas-FtWorth\n\
Detroit\n\
Omaha\n\
Phoenix\n\
SaltLake");

rather than how I did it:

```
 void sec1L_List(Bar& bar) {
  bar.Add("Billings", THISBACK(s1L));
  bar.Add("Cincinnati", THISBACK(s1L));
  bar.Add("ElPaso", THISBACK(s1L));
  bar.Add("GreenBay", THISBACK(s1L));
  bar.Add("Houston", THISBACK(s1L));
  bar.Add("Jacksonville", THISBACK(s1L));
  bar.Add("KansasCity", THISBACK(s1L));
  bar.Add("Montreal", THISBACK(s1L));
  bar.Add("Houston", THISBACK(s1L));
  bar.Add("SanFrancisco", THISBACK(s1L));
  bar.Add("Seattle", THISBACK(s1L));
          bar.Separator();
  bar.Add("Fairbanks", THISBACK(s1L));
  bar.Add("Seward", THISBACK(s1L));
 }
```

With my code it was necessary to click on one of the names in the "text box".

With "label.SetLabel" and "bar.AddNC(label)" I had to have a THISBACK in the text box and that is triggered with a mouse over rather than a click.

I would like to be able to click on 1L.

It may be simple but I don't understand how or why.
Attached is a modification of your code with a .upp file.

Neil

Subject: Re: menu Tree
Posted by Sender Ghost on Sun, 15 May 2011 17:36:21 GMT
View Forum Message <> Reply to Message

Well, with current implementation of MenuBar, I think, it also possible to do as follows:

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class App : public TopWindow {
private:
 bool tac1L, tac1R;
public:
 typedef App CLASSNAME;
 App();
 // Ctrls
 MenuBar menu;
 Label label_1L, label_1R;
 // Bars
 void MainBar(Bar& bar);
 void FileBar(Bar& bar);
 void TacMenu1L(Bar& bar);
 void TacMenu1R(Bar& bar);
 // Events
 void OnCheck1L();
 void OnCheck1R();
};

App::App() : tac1L(false), tac1R(true) {
 Title("MenuBar test application");
 CenterScreen().Sizeable().MinimizeBox().MaximizeBox();
 SetRect(Size(320, 240));
```

```cpp
	AddFrame(menu);
	AddFrame(TopSeparatorFrame());
	menu.Set(THISBACK(MainBar));

	label_1L.SetLabel("Atlanta\nBrownsville\nDallas-FtWorth\nDetroit\nOmaha\nPhoenix\nSaltLake")
.NoWantFocus();
	label_1R.SetLabel("Other cities\n.\n..\n...").NoWantFocus();
}
// Bars
void App::MainBar(Bar& bar){
	bar.Add("File", THISBACK(FileBar));
}

void App::FileBar(Bar& bar){
	bar.Add("1L", THISBACK(TacMenu1L)).Check(tac1L);
	bar.Add("1R", THISBACK(TacMenu1R)).Check(tac1R);
	bar.Separator();
	bar.Add("Exit", THISBACK(Close)).Key(K_CTRL_Q);
}

void App::TacMenu1L(Bar& bar) {
	bar.Add(tac1L ? "Un&select" : "&Select", THISBACK(OnCheck1L)).Image(tac1L ? Null :
CtrlImg::smallcheck());
	bar.Separator();
	bar.AddNC(label_1L);
}

void App::TacMenu1R(Bar& bar)
{
	bar.Add(tac1R ? "Un&select" : "&Select", THISBACK(OnCheck1R)).Image(tac1R ? Null :
CtrlImg::smallcheck());
	bar.Separator();
	bar.AddNC(label_1R);
}
// Events
void App::OnCheck1L()
{
	tac1L = !tac1L;
}

void App::OnCheck1R()
{
	tac1R = !tac1R;
}

GUI_APP_MAIN
{
	App app;
```

```
 app.Run();
}
```

Because, the checkbox, which you "would like to be able to click", is clicked, when you selected it to show next menu bar (according to source code, which you wrote).
Also, the same Label Ctrl, showed on the current menu bar, couldn't be added to the next bar (because it will be "moved", in this case). Instead, try to use specific Label Ctrl for specific menu bar.

File Attachments
1) MenuBarTest.png, downloaded 805 times

Subject: Re: menu Tree
Posted by nlneilson on Sun, 15 May 2011 23:54:59 GMT
View Forum Message <> Reply to Message

Thanks Sender Ghost

Sender Ghost wrote on Sun, 15 May 2011 19:361. Well, with current implementation of MenuBar, I think, it also possible to do as follows:

2. Because, the checkbox, which you "would like to be able to click", is clicked, when you selected it to show next menu bar (according to source code, which you wrote).

3. Also, the same Label Ctrl, showed on the current menu bar, couldn't be added to the next bar (because it will be "moved", in this case). Instead, try to use specific Label Ctrl for specific menu bar.

2. That is where I am confused. To show the "text box" it is just a mouse over and not an actual click.

As in the image above (message #32382) that is what is needed so a mouse over 1L, 1R, 2L and 2R will show the names included in one of those 4 layers to display in a Java app (NASA WWJ) the Upp app interacts with.

Your latest code makes it so an actual click is required on Select/UnSelect for one of those 4 layers similar to any of the names in my app had to be clicked. My first mod of your code the mouse over enabled it rather than a click, couldn't figure out why which made my head hurt.

3. I had not started working on the second Label since I could not get the first to work correctly.

Your coding style is different from mine. You declare the function/method first. I do that in header files but not in my main since it is easier to add/remove when coding/testing and when I get something working I don't bother to change it.

I seldom use "tac1R ? "Un&select" : "&Select",..." instead of if blocks because it's easier to put a Break Point for debugging (and for me to follow) when they are nested several deep.

1.   "... with current implementation of MenuBar ..."
Apparently with your code "Select" needs to be clicked and in my code any of the names need to be clicked.

Clicking on "1L" will not work.

A user should be able to get used to that.

Thanks for the help!

Neil

Subject: Re: menu Tree
Posted by Sender Ghost on Mon, 16 May 2011 01:54:41 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Mon, 16 May 2011 01:54
Your coding style is different from mine. You declare the function/method first. I do that in header files but not in my main since it is easier to add/remove when coding/testing and when I get something working I don't bother to change it.

I also write function declarations inside header files and their implementations inside C/C++ files. It presented in simplified form here, without referring to specific names of header files.
nlneilson wrote on Mon, 16 May 2011 01:54
Clicking on "1L" will not work.

A user should be able to get used to that.

When you use Check(bool check) method of Bar, it uses check variable to show checked state of passed variable to Check method (e.g. tac1L, tac1R). It is not an Option Ctrl, rather some kind of drawing of CtrlImg::MenuCheck0(), CtrlImg::MenuCheck1(), etc. images, depending from state.

I suggested to use Label Ctrl to show "informative" text, instead of current bar.Separator(); without such a text (e.g. some named sections of menu bar). Currently, we (you and me) overused it, I think. For example, it also possible to create a Label Ctrl, which can be clicked, but this didn't change the current implementation of MenuBar, where Check method is not an Option Ctrl, paraphrasing.

Now, I suggest to use MenuBar just to show current checked values and change them on one-click basis. For more detailed interface, you can create "Settings" dialog window with more detailed options, which you can access from MenuBar, ToolBar, etc.

Thanks for your attention, too.

Sender Ghost wrote on Mon, 16 May 2011 03:54
I also write function declarations inside header files and their implementations inside C/C++ files.

Someone mentioned having the implementations in a header file could cause problems, I do but have not had problems yet, something to look forward to.

Quote:When you use Check(bool check) method of Bar, it uses check variable to show checked state of passed variable to Check method (e.g. tac1L, tac1R). It is not an Option Ctrl, rather some kind of drawing of CtrlImg::MenuCheck0(), CtrlImg::MenuCheck1(), etc. images, depending from state.

That worked OK, is that just drawing an image of the box with/without the check mark you are pointing out?

Quote:I suggested to use Label Ctrl to show "informative" text, instead of current bar.Separator(); without such a text (e.g. some named sections of menu bar). Currently, we (you and me) overused it, I think. For example, it also possible to create a Label Ctrl, which can be clicked, but this didn't change the current implementation of MenuBar, where Check method is not an Option Ctrl, paraphrasing.

I didn't want the label text to be clickable so that worked OK.

Quote:Now, I suggest to use MenuBar just to show current checked values and change them on one-click basis. For more detailed interface, you can create "Settings" dialog window with more detailed options, which you can access from MenuBar, ToolBar, etc.

Is this what you had in your previous code or are you suggesting something different?

With your Label and bar.AddNC(label) example I was able to add the text "- Distance -", "- Angle -" and "- dec Places -" in my Settings menu. I had this in Java but after porting to C++/Upp I was unable to do that before.

I appreciate the help.

Neil

edit: Is there an easy way to set the text in the label bold?

Subject: Re: menu Tree
Posted by Sender Ghost on Mon, 16 May 2011 16:50:10 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Mon, 16 May 2011 10:31
That worked OK, is that just drawing an image of the box with/without the check mark you are pointing out?

Yes, but with internal implementation. External implementation could look like follows:

```
void App::FileBar(Bar& bar) {
 bar.Add("1L", THISBACK(OnCheck1L)).Image(tac1L ? CtrlImg::MenuCheck1() :
CtrlImg::MenuCheck0());
}
```

nlneilson wrote on Mon, 16 May 2011 10:31
Is this what you had in your previous code or are you suggesting something different?

Something different: New dialog window with more detailed settings, without using MenuBar. It depends from your implementation. And using "MenuBar just to show current checked values and change them on one-click basis" on main window.
nlneilson wrote on Mon, 16 May 2011 10:31
Is there an easy way to set the text in the label bold?

By using QTF after \1 escape byte:

Label text;
text.SetLabel("\1[*1 Some informative text]");

or through Font settings:

Label text;
text.SetLabel("Some informative text").SetFont(text.GetFont().Bold());

Subject: Re: menu Tree
Posted by nlneilson on Tue, 17 May 2011 01:51:37 GMT
View Forum Message <> Reply to Message

Thanks for the QTF link and the line of code showing how it is done. The bold text works good.

Sender Ghost wrote on Mon, 16 May 2011 18:50
Something different: New dialog window with more detailed settings, without using MenuBar. It depends from your implementation. And using "MenuBar just to show current checked values and change them on one-click basis" on main window.

This I don't understand.

"... without using MenuBar" --- "And using "MenuBar ..."

Subject: Re: menu Tree
Posted by Sender Ghost on Tue, 17 May 2011 02:42:02 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Tue, 17 May 2011 03:51
This I don't understand.

"... without using MenuBar" --- "And using "MenuBar ..."

Just read full context. In simplified form it looks like follows:

```cpp
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class Window : public TopWindow {
public:
 typedef Window CLASSNAME;
 Window();

 ArrayCtrl list;
};

Window::Window()
{
 Title("New Window");
 SetRect(Size(320, 240));

 list.HSizePosZ(4, 4).VSizePosZ(4, 4);
 list.AddColumn(String());

 for (int i = 0; i < 10; ++i)
 {
  list.Add(AsString(i + 1));
 }

 Add(list);
}

class App : public TopWindow {
public:
```

```
    typedef App CLASSNAME;
    App();

    MenuBar menu;
    Window window;

    void MainBar(Bar& bar);
    void FileBar(Bar& bar);
    void OnSettings();
};

App::App()
{
    Title("MenuBar test application");
    SetRect(Size(640, 480));
    AddFrame(menu);
    AddFrame(TopSeparatorFrame());
    menu.Set(THISBACK(MainBar));
}

void App::MainBar(Bar& bar)
{
    bar.Add("File", THISBACK(FileBar));
}

void App::FileBar(Bar& bar)
{
    bar.Add("Settings", THISBACK(OnSettings));
    bar.Separator();
    bar.Add("Exit", THISBACK(Close));
}

void App::OnSettings()
{
    window.Execute();
}

GUI_APP_MAIN
{
    App app;
    app.Run();
}
```

I guess, this is not what you looking for, but it was that I suggested.

You can continue to find exact solution, of course. E.g. trying to extend MenuBar to support
Option with right check (>), which shows next menu bar, after mouseover on it (mimic MenuItem).

```
// Somewhere inside derived TopWindow class
Option opt;
// Then add it for some Bar
bar.AddNC(opt);
```

Or using different solutions, not mentioned here.

Subject: Re: menu Tree
Posted by nlneilson on Tue, 17 May 2011 04:03:14 GMT
View Forum Message <> Reply to Message

Thanks again for the code example.

I could add that bar to show all the layers that are enabled.

This is what I use to show the last layer picked.
In Java I renamed the "Distance" label to "Layer" for that text field and when calculating a geodesic distance it would change back to "Distance".
I think I tried a Upp app that does this but have not implemented it in this one yet.

What your code shows is what I tried for a "Help" sub menu, until my head hurt. Now I should be able to do it.

A question on that is can I pull up a text file into that new window?

Subject: Re: menu Tree
Posted by Sender Ghost on Tue, 17 May 2011 14:43:21 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Tue, 17 May 2011 06:03
A question on that is can I pull up a text file into that new window?

Yes, just add appropriate Ctrl for this window and access it from main window.

```
App::App()
{
 //..
 String data = LoadFile("path/to/file");
 // text here is added Ctrl;
 window.text.SetData(data);
}
```

Subject: Re: menu Tree
Posted by nlneilson on Tue, 17 May 2011 19:57:54 GMT

View Forum Message <> Reply to Message

I had a problem pulling a file in that way.
I will tinker with that later for a short help file.

What I did before was have help files a user could put an Icon link on his OS screen in .txt, .pdf or
.html. The .html worked the best and was much easier to link to different sections than a .pdf and
could be opened with any browser.

Thanks for the help getting informative text in the menu and I will eventually get the extra window
implemented.