Subject: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 15 Sep 2007 14:48:58 GMT
View Forum Message <> Reply to Message

Hi,

I've been using the u++ for almost two months, and got very fond of it. I am currently working on a dockctrl widget which would act as a dockable window (like those you find in the VS). Although currently it is under heavy development, I would like to share my dockctrl project's progress here.

Edit: Message Body and history removed.

ChangeLog: (Only the newest version. You can find full changelog in the DockCtrl SVN repository)

0.52
> public release name DockCtrl DEV803b.1
---
3/21/2008
+ added full serialization support.
+ added default/custom layout support.
* DockableCtrl class reworked (now it is more abstract);
* TabWindow class reworked (no longer uses PaneSplitter() class to control child widgets).
* DockCtrl class renamed to DockBase class and added a more generic DockCtrl class.
* DockBase class derives from TopWindow.
+ added InitDockCtrl() method for widget initialization.
+ added InitCustomLayouts() method for planned "predefined" layout support.
+ added Tabify() method (with 1 overload) to simplify widget tabbing.
+ added DockFloat() method to DockableCtrl class.
+ added ShowDragBarButtons() method to DockWindow class.
+ added ForbidDragAndDrop()/PermitDragAndDrop() methods to DockableCtrl base widget class.
dnd behaviour of each DockableCtrl derived widget can be controlled using this method pair.
+ added layout group to the settings window.
+ added HotKey support for Settings window (default is CTRL + HOME).
+ added methods to control forbid/permit on window sides.
- numerous bugfixes, and improvements.

3/22/2008
+ double clicking on a dragbar of a authidden dockwindow will cause it go into docked state.
+ added "stdsize" variable to the PaneFrame class.
+ added ShowXXXXPane() (X:left/top/right/bottom) public methods to the Dockbase class.

0.52
> public release name DockCtrl DEV803b.2
---

3/27/2008

+ added widget grouping (both predefined/default and user) support.
+ added WidgetGroup() method and defined "%" operator for widget grouping.
+ added group manager to the settings window.
* tab window creation/managment mechanism changed.

To Do:
---
- add a self-explanatory "Complex Dockable Framework" example to the package.

- add drag-n-drop animations.
- add DockBar class -- a dockable Toolbar for DockCtrl framework
- write documentation.
- fix possible major, and minor bugs.
- cleaning up the code.
- bla, bla...

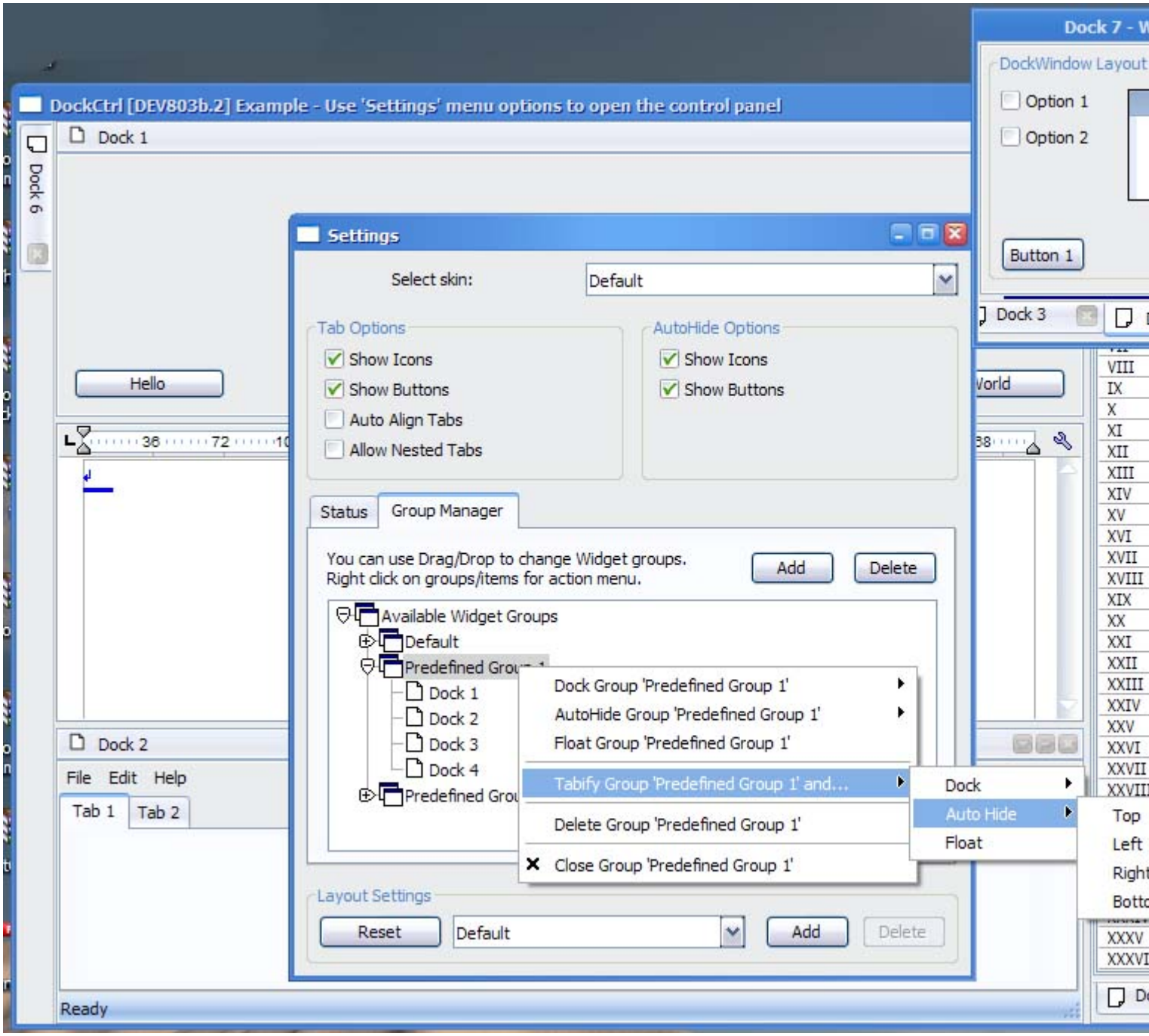
Any suggestions, bug reports and constructive criticism are always welcome.

You can find the source code of the DockCtrl framework example here or the latest build in the the U++ SVN/Bazaar folder.

Regards.

## File Attachments

1) DockCtrl Example Exe (DEV803b.2).rar, downloaded 640 times
2) DockCtrl DEV803b.2.jpg, downloaded 2776 times




3) DockCtrl Package DEV(803b.2).rar, downloaded 615 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Sat, 15 Sep 2007 14:58:07 GMT
View Forum Message <> Reply to Message

Promising.

Mirek

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mezise on Sat, 15 Sep 2007 15:39:23 GMT
View Forum Message <> Reply to Message

Screenshots look promising

Personally I vote for expanding docking system to Perspectives idea of Eclipse IDE.

http://www.eclipse.org/articles/using-perspectives/Perspecti veArticle.html
http://aptana.com/docs/index.php/About_the_Eclipse_Workbench

Perspectives together with:
- Editors area
- Views areas
- Tabs that may be in Maximize, Fast View, Detached modes
are very convenient and efficient.

I love these features of Eclipse.

Michal

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 15 Sep 2007 16:25:38 GMT
View Forum Message <> Reply to Message

Thank you

Well, it shouldn't be so difficult to implement some features of the "perspectives" of the Eclipse. In fact, I was considering to adopt some of the features of it, when I first started this DockCtrl project. Some of them are easy to adopt due to the flexible framework of the U++. But others will be extremely difficult to implement, for the "perspectives" is AFAIK not a widget It is some sort of a mature, layout/GUI framework. So, It is well beyond the scope of this project (for a while, at least). The scope of this project to provide some flexible and reusable GUI widget). Sure, the dockctrl will be and (already is) customizable.
And if you could be more specific instead of suggesting me to implement a whole framework , I will add it to my TODO list, if it is not beyond my coding limits.

regards.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Sat, 15 Sep 2007 16:48:16 GMT
View Forum Message <> Reply to Message

Quote:As I stated above, it is still in alpha state, so at this point i cannot provide you with the working example (the reason is, besides it being under heavy development, the source code is very messy; thus needs refactoring and cleanup).

Excellent job! Could you post here executable (win32) version?

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 18 Sep 2007 12:01:57 GMT
View Forum Message <> Reply to Message

unodgs wrote on Sat, 15 September 2007 19:48Quote:As I stated above, it is still in alpha state, so at this point i cannot provide you with the working example (the reason is, besides it being under heavy development, the source code is very messy; thus needs refactoring and cleanup).
Excellent job! Could you post here executable (win32) version?

Thank you and of course I can post it.
And I'm sorry for the delay I've been very busy. But please keep in mind that this example is only a sort of "proof-of-concept" of that it is possible to implement this type of control without using any external lib. It depends %100 on U++ widgets. So that it is more a "widget container" than a widget. Good news is it will probably take a month or so to implement a full featured dockctrl.

By the way, there was a bug in smooth slide function of hide/autohide option which caused dockctrl to crash (due to my awful coding stlye ) I've spotted it but had no time to fix it, so I disabled hide/autohide feature. I will fix it probably in a week.

Edit: executable changed. Found 2 bugs and fixed.
Edit: executable moved to the first post.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Tue, 18 Sep 2007 12:31:14 GMT
View Forum Message <> Reply to Message

Wow! Very nice! Only drag and drop is missing and docking to tab control but you need Mirek's cooperation here. As for DnD please download Qt 4.3.1 and see similar example. I like very much what they did. Anyway, keep going!

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Tue, 18 Sep 2007 18:35:10 GMT
View Forum Message <> Reply to Message

I was wrong about tabctrl widget. It should not be tauched. Instead your DockCtrl should be able to draw tabs if one pane is put over second (sorry if this is not in english ) and be also able to draw those tabs at bottom (maybe from the left and right side too). Anyway your control should do all that things itself.

Subject: Re: DockCtrl (A dockable window widget for U++)

Posted by unodgs on Tue, 18 Sep 2007 18:40:49 GMT
View Forum Message <> Reply to Message

I also copied mainwidnow.exe example from qt on upp public ftp. Look for upp/public/!mainwindow!.7z Of course you don't have to follow their way. I only wanted to show you one complete solution.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 18 Sep 2007 19:05:14 GMT
View Forum Message <> Reply to Message

unodgs wrote on Tue, 18 September 2007 21:40I also copied mainwidnow.exe example from qt on upp public ftp. Look for upp/public/!mainwindow!.7z Of course you don't have to follow their way. I only wanted to show you one complete solution.

Ok, I will look for it and will examine the qt example for sure. As I said before, I am open to all suggestions.
Also, I will probably update this small demo-like example per-week. You will be able find the updated exe in my first post. Thank you for your help and feedbacks.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 19 Sep 2007 11:04:23 GMT
View Forum Message <> Reply to Message

unodgs wrote on Tue, 18 September 2007 21:35I was wrong about tabctrl widget. It should not be tauched. Instead your DockCtrl should be able to draw tabs if one pane is put over second (sorry if this is not in english ) and be also able to draw those tabs at bottom (maybe from the left and right side too). Anyway your control should do all that things itself.

I've examined the qt's dockwidget and understood what you mean with the tabbed docks. In my opinion, it shouldn't be very difficult to implement such a feature. And I have a good news:

Last night I've cleaned up (and translated to english) the code and redesigned the DockCtrl core classes. Now every dockctrl "client" (conceptually, DockCtrl is some sort of a "server") is derived from a base class called DockableCtrl. So the DockCtrl is much more flexible and is possible from now on to add/create new dockable widgets like dockbars (which are similar to windows ReBars), or others. Once I complete the DockWindow implementation, I will go for a DockBar (actually it is piece of a cake to implement one, thank to U++ and all you developers). If everything goes well, I may be able to publish the source code in a month or so.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 19 Sep 2007 12:27:24 GMT

Oblivion wrote on Wed, 19 September 2007 07:04
Last night I've cleaned up (and translated to english) the code and redesigned the DockCtrl core classes. Now every dockctrl "client" (conceptually, DockCtrl is some sort of a "server") is derived from a base class called DockableCtrl. So the DockCtrl is much more flexible and is possible from now on to add/create new dockable widgets like dockbars (which are similar to windows ReBars), or others. Once I complete the DockWindow implementation, I will go for a DockBar (actually it is piece of a cake to implement one, thank to U++ and all you developers). If everything goes well, I may be able to publish the source code in a month or so.
Wonderful! I'm sure we will use it for theide as soon as you finish it (maybe some little fixes (if any) would have to be done by us(us = Mirek ). Personaly I can't wait to see it in action

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 26 Sep 2007 12:57:59 GMT


Ok, from now on DockCtrl has its very basic DnD mechanism (no position/placement awareness added yet). I've posted the new version of the example exe. You can test it

I am trying to implement two types of DnD mechanism. First one uses a sort of tracker rectangle (which is, at the moment for testing-purpose only (so please don't get dissapointed by it's ugliness, it will be refined as well ). It works fine. Second "will" use solid window dragging (as in QT). But the problem is solid window draging needs a neat trick which will involve win32 API functions, so I will need some assistance about X11 equivalents. In fact, DockCtrl already uses some win32 API calls, so I need help . Here are the code snippets which need to be implemented in X11:
(don't get confused about the other methods, I only need X11 API calls)

```
//=================================================
#ifdef PLATFORM_WIN32
LRESULT  DockWindow::WindowProc(UINT message, WPARAM wParam, LPARAM lParam)
 {
  UINT defmsg  = TopWindow::WindowProc(message, wParam, lParam);

  if(IsFloating())
  {
   if(message == WM_MOVING)
   {
    POINT pos; GetCursorPos(&pos);
    DockCtrlMessage(DMSG_DRAGGING, Point(pos.x, pos.y));
   }
   if(defmsg == HTCAPTION)
   {
     if(DnDHasTarget())
     {
```

```
    Dock(DndGetTarget(), State(), Placement(), Position());
    Refresh();
    DnDTargetORange();
   }

  }
 }

 return defmsg;
}
#endif
  //=============================================================

Rect DockableCtrl::GetWindowRect(TopWindow& window, Rect& r)
{
#ifdef PLATFORM_WIN32

 Rect wrect = r;
 ::AdjustWindowRect(wrect, window.GetStyle(), false); //Win32 call
 return wrect;

#endif

  //=============================================================
```

Any suggestions, help and code are welcome

Regards

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 26 Sep 2007 19:49:41 GMT
View Forum Message <> Reply to Message

Hi! Glad to see DnD! Agree that xor rectangle is ugly and what's more - under vista it moves very slow. But as you pointed it's only to test. My advice is - finish it with native api if you need. Then we try to extend upp interface in a proper way.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 26 Sep 2007 20:20:33 GMT
View Forum Message <> Reply to Message

unodgs wrote on Wed, 26 September 2007 22:49Hi! Glad to see DnD! Agree that xor rectangle is ugly and what's more - under vista it moves very slow. But as you pointed it's only to test. My advice is - finish it with native api if you need. Then we try to extend upp interface in a proper way.

Hmm, interesting. I tested the example on Win98se (notebook), /XP(SP2) and Vista Basic (desktop) and it worked fine. Do you use Aero? Maybe it could be the reason or there may be a driver issue??. Anyway, it will be removed when I find a solution to the solid window dragging (I already implemented but then disabled it because there were too much flickering).

Regards

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Fri, 28 Sep 2007 10:58:28 GMT
View Forum Message <> Reply to Message

Quote:Hmm, interesting. I tested the example on Win98se (notebook), /XP(SP2) and Vista Basic (desktop) and it worked fine. Do you use Aero? Maybe it could be the reason or there may be a driver issue??. Anyway, it will be removed when I find a solution to the solid window dragging (I already implemented but then disabled it because there were too much flickering).
Yes, I used Aero. I check on xp and everything was OK. I was thinking about flickering as we should support both visual styles of dragging windows (full and broder only). You must use "smarter" way of painting rectangle. I'll try to investigate in this topic. Or you can make a rectangle thiner (it can be even one pixel width).

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 13:16:12 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Wed, 26 September 2007 08:57
```
Rect DockableCtrl::GetWindowRect(TopWindow& window, Rect& r)
{
#ifdef PLATFORM_WIN32

 Rect wrect = r;
 ::AdjustWindowRect(wrect, window.GetStyle(), false); //Win32 call
 return wrect;

#endif
```

BTW, are you aware that in U++, rect of window excludes "decorations"?

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 13:18:41 GMT
View Forum Message <> Reply to Message

unodgs wrote on Wed, 26 September 2007 15:49Hi! Glad to see DnD! Agree that xor rectangle is

ugly and what's more - under vista it moves very slow. But as you pointed it's only to test. My advice is - finish it with native api if you need. Then we try to extend upp interface in a proper way.

Interesting, what xor rectangle?

BTW, are you aware about

void DrawDragRect(Draw& w, const Rect& _rect1, const Rect& _rect2, const Rect& _clip, int n, Color color, uint64 pattern)

At least, that solves the flickering problem...

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Fri, 28 Sep 2007 14:00:55 GMT
View Forum Message <> Reply to Message

Quote:Interesting, what xor rectangle?
Xor works perfectly if border of rectangle is 1 pixel width. If it is more than 1 pixel and you move rectnagle by 1 pixel for example rectangle must flicker - overlaping border parts are deleted and then repainted in the same position.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 14:49:02 GMT
View Forum Message <> Reply to Message

Well, tracker (xor) rectangle implementation is suspended for a while. Good news is, I've managed to add solid-window dragging on win32. (As usual, I've uploaded it; you can find it on the first post) It works fine on my OS configurations (win98/XP(sp2)/Vista basic), except for Win98. AFAIK W98 doesn't support solid window moving by default; it uses standard tracker rectangle of the system. bad news is that I've managed solid dragging by using win32 api. So it is win32 spesific. But it is nested in #ifdef/endif preprocessor commands and is not very lenghty (at most, 100 lines of code), so it should be easy to implement it in X11 (or I hope so )

The main problem I've encoutered was to implement a workaround for famous windows "feature:" WM_NCLBUTTONUP. IT is explicitly stated in the MSDN that this message is sent to the window after left button released. But guess what, this is definitely wrong on XP or greater! Usual microsoft behaviour: "It's not a bug, it's a feature!"...
So I had to implement an aplication wide mouse hook.

By the way, Mirek why isn't there any "non-client area" event handler on U++? I couldn't find any. Did I miss something? Imho, at least the Topwindow class should have protected virtual methods

for these messages. And as for the U++ documentation... Well, U++ is like an Iceberg, almost everything is gone undocumented. Don't get me wrong, The actual U++ code is impressing; but lack of documentation really slows down the process, and maybe that is why some people think that it is only a gui widget collection (well, it is definitely not!)

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 14:59:48 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 28 September 2007 16:16Oblivion wrote on Wed, 26 September 2007 08:57
Rect DockableCtrl::GetWindowRect(TopWindow& window, Rect& r)
{
#ifdef PLATFORM_WIN32

 Rect wrect = r;
 ::AdjustWindowRect(wrect, window.GetStyle(), false); //Win32 call
 return wrect;

#endif

BTW, are you aware that in U++, rect of window excludes "decorations"?

Yes I am well aware of that. And that caused me some trouble
But are there any "rectangle" methods that doesn't exclude them? I hate to take care of nc area manually all the time

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 16:24:36 GMT
View Forum Message <> Reply to Message

unodgs wrote on Fri, 28 September 2007 10:00Quote:Interesting, what xor rectangle?
Xor works perfectly if border of rectangle is 1 pixel width. If it is more than 1 pixel and you move rectnagle by 1 pixel for example rectangle must flicker - overlaping border parts are deleted and then repainted in the same position.

AFAIK not with

Quote:
void DrawDragRect(Draw& w, const Rect& _rect1, const Rect& _rect2, const Rect& _clip, int n, Color color, uint64 pattern)

Search it in CtrlCore for usage, this should be flicker-free - it works by xoring only "new" parts of

rectangle - overlapping parts are NOT deleted

Mirek

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 16:27:15 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 28 September 2007 10:59
Yes I am well aware of that. And that caused me some trouble
But are there any "rectangle" methods that doesn't exclude them? I hate to take care of nc area manually all the time

Well, the important reason for this is that in X11, there is generally not possible to use reactangle with decorations. X11 only allows you to work with the "client" rectangle. That is why all such methods were carefully removed from U++ so that we are X11 compatible....

I am afraid you will have to find a way not using AdjustRect if you want this to work in Linux.

Mirek

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 16:38:45 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 28 September 2007 10:49
By the way, Mirek why isn't there any "non-client area" event handler on U++? I couldn't find any. Did I miss something?

WindowProc?

Anyway, I am starting to feel a bit worried about all these platform specific tricks. I am not sure what exactly is covered by "docking windows", but I expected that most of it can be done using existing interfaces.

Quote:
Well, U++ is like an Iceberg, almost everything is gone undocumented. Don't get me wrong, The actual U++ code is impressing; but lack of documentation really slows down the process, and maybe that is why some people think that it is only a gui widget collection (well, it is definitely not!)

Yes. You can start improving the situation by writing some docs

I am starting planning for the next year, theide / documentation will be the main focus.

Mirek

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 16:57:32 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 28 September 2007 19:38Oblivion wrote on Fri, 28 September 2007 10:49
By the way, Mirek why isn't there any "non-client area" event handler on U++? I couldn't find any. Did I miss something?

WindowProc?

Anyway, I am starting to feel a bit worried about all these platform specific tricks. I am not sure what exactly is covered by "docking windows", but I expected that most of it can be done using existing interfaces.

Quote:
Well, U++ is like an Iceberg, almost everything is gone undocumented. Don't get me wrong, The actual U++ code is impressing; but lack of documentation really slows down the process, and maybe that is why some people think that it is only a gui widget collection (well, it is definitely not!)

Yes. You can start improving the situation by writing some docs

I am starting planning for the next year, theide / documentation will be the main focus.

Mirek

YEs there is Windowproc, and I use it. What I meant with "nc handlers" was, methods to handle windowproc messages as in MFC. Anyway, it was only a suggestion

Well I think I exaggerated the situation a bit. I've managed to handle all those "tricks" I've mentioned before (including AjustWindowRect) in native U++. So you should'nt have to worry about them anymore  Only the "Mousehook" remains. It only handles WM_NCLBUTTONUP, and is needed because windows has a well known internal bug which they prefer to call "feature". So there is nothing much left to worry about. As I'd stated in the first post, it is and will be in U++ (except for the mouse hook). IMO, all those tricky things which I tried to implement couple of days ago are due to the lack of documentation (it takes time to find what i need).

Actually, I am working on documentation for DockCtrl. But my english, as you may have noticed, is far from being perfect  And this is my "internal bug"

Regards

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 18:59:21 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 28 September 2007 12:57
It only
handles WM_NCLBUTTONUP, and is needed because windows has a well known internal bug which they prefer to call "feature".

Well, what is your need for WM_NCLBUTTONUP and what is the bug?

I believe that if there would be any problem, we should have noticed...

Quote:
Actually, I am working on documentation for DockCtrl. But my english, as you may have noticed, is far from being perfect  And this is my "internal bug"

Regards

As is mine. The content, at this phase, is much more imporant.

Mirek

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 19:40:19 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 28 September 2007 21:59

Well, what is your need for WM_NCLBUTTONUP and what is the bug?

Ok, let me explain. Dockwindows are basically derived from "DockableCtrl" class, which is actually a derivative of TopWindow. So it is possible to use it at the same time as a ctrl, and  as a window. In their "docked state" they act as a ctrl and in their "float state" they are detached (dragged/teared) from their panes and activated as tool windows. When a dockwindow is teared off/dragged in solid window dragging(with the Dockablectrl::Float()/FloatEx method), it has to receive mouse messages immediately, because it is from then on in a "dragging state". This can be done easily by sending a fake WM_NCLBUTTONDOWN message with HTCAPTION parameter; by this command, system is faked to believe that mouse is dragging a "window" from

it's titlebar. Here is the sequence.

1. Drag Dock from its "dragbar"(from the title bar of the dock)
   This is done by LeftDrag(). So the left mouse is, from now on, "down"
2. Remove() dock from its pane.
3. Open it as a child toolwindow under the cursor position.
4. Immediately Send a WM_NCLBUTTONDOWN message explicitly (with HTCAPTION as wparam). (because, the Lmousebutton is still down.)
5. Move window.
6. Drop (theoretically, the window should send WM_NCLBUTTONUP so we could get the drag/drop state, but it doesnt; so we use the mouse patch we already installed when we invoke the Float() command and send the WM_NCLBUTTONUP command manually).

This is the actual tear-off/drag sequence.

(As you can see this in the executable example (0.49.7a) I've posted. This is how a solid window drag sequence can be made possible on Win32, Or maybe I don't know the way,  ayn suggestions will be very useful). But unfortunately windows from XP on does not send the proper WM_NCLBUTTONUP message to the window's message pump when the left button is released on the NC area! So it couldn't "drop" at first time when mouse button is released. It is clearly stated in the MSDN that it does, but in fact it doesn't. It is a well known bug/behaviour.

So as you see, in this case the standard U++ leftup or leftdown won't work because those messages only handle AFAIK the client area. Is there something I don't know about it?

Here is the problem and the hook solution is explained:

http://www.codeguru.com/cpp/misc/misc/windowsmessaging/artic le.php/c3885/

I use a modified version of this mouse hook, So the article could give you a clear idea about the problem, solution and (hopefully)  a better solution suggestion from you

Here is the modified code

```
//=====================================================
// Original code by Robert Wiejak, (c) 2001.
//
//
#ifdef PLATFORM_WIN32
LRESULT DockCtrl::Win32MessagePatch(int code, WPARAM wParam, LPARAM lParam)
{

 if(code == 0)
 {
  PMOUSEHOOKSTRUCT mousehookstruct = (PMOUSEHOOKSTRUCT) lParam;

  DockableCtrl* dock = ::__dctrlptr->ptr->FindDockWindow(mousehookstruct->hwnd);
  {
```

```
   switch(wParam)
   {
    case WM_NCLBUTTONDOWN:
      if(dock && mousehookstruct->wHitTestCode == HTCAPTION) dock->Dragging(TRUE);
     break;

    case WM_NCLBUTTONUP:

      if(dock) dock->Dragging(FALSE);
     break;

    case WM_LBUTTONUP:
      {
       DockableCtrl *target = NULL;
       if(dock && dock->IsDragged()) target = dock;
       else
       target = ::__dctrlptr->ptr->FindDraggedWindow();

       if(target)
       {
        ::PostMessage(target->GetHWND(), WM_NCLBUTTONUP, HTCAPTION,
MAKELONG(mousehookstruct->pt.x, mousehookstruct->pt.y));
        target->Dragging(FALSE);
       }
      }
     break;

    default:
     break;
   }
  }
 }
//================================================
```
Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Fri, 28 Sep 2007 20:09:55 GMT
View Forum Message <> Reply to Message

I see, thanks.

I think you can find this article useful:

http://slingkid.blogsome.com/

Quote:
The trick in asking the OS to resize a window is sending the window a WM_SYSCOMMAND message. The following are parameters you pass in the message to specify how the window should be resized.

```
public const int SC_DRAGMOVE = 0xF012;
public const int SC_DRAGSIZE_N = 0xF003;
public const int SC_DRAGSIZE_S = 0xF006;
public const int SC_DRAGSIZE_E = 0xF002;
public const int SC_DRAGSIZE_W = 0xF001;
public const int SC_DRAGSIZE_NW = 0xF004;
public const int SC_DRAGSIZE_NE = 0xF005;
public const int SC_DRAGSIZE_SW = 0xF007;
public const int SC_DRAGSIZE_SE = 0xF008;
```

Usually, in your OnMouseDown method, you would delegate the job of resizing to ths OS. The code below does just that.

We are already using this in U++ - look at

```
void SizeGrip::LeftDown(Point p, dword flags)
```

It seems to work well on all platforms.

I believe that using SC_DRAGMOVE = 0xF012 instead of 0xf008 should solve your problem...

In the same function, there is also X11 implementation, based on

http://standards.freedesktop.org/wm-spec/1.3/ar01s04.html

As you can see, replacing

```
  m.data.l[2] = 8;
```

should do the trick for X11...

Mirek

(BTW, it perhaps might be useful to make this more public in CtrlCore).

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 20:13:58 GMT
View Forum Message <> Reply to Message

Thank you very much Mirek. I hope that I will get rid of the hook this week with the inof you've provided.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 22:09:01 GMT
View Forum Message <> Reply to Message

With your hint, I removed all the non-U++ code from dockctrl. There is no platform-specific trick left, and guess what: It works perfectly  I owe you a " U++ documentation" Mirek. (Or would you prefer a beer?  )

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Fri, 28 Sep 2007 22:15:25 GMT
View Forum Message <> Reply to Message

I've been working on something similar (I've been off the net for two weeks or I would have said something sooner), and I'm sending  the WM_NCLBUTTONDOWN message in the same way. I get around it like this:

```
if (message == WM_WINDOWPOSCHANGING && !IsDocked() && !GetMouseLeft())
   // Dragging finished
```

(IsDocked() is mine, GetMouseLeft() is UPP)
Fortunately this is almost the only hack I've had to use, the rest is all UPP.

I've attached a demo compiled with MingW (which seems much less stable than the MSC version, annoyingly). As you can see I've still got some considerable bugs to work out myself  I've only just managed to get autohide mostly working but the Splitters need some serious attention to get them to resize correctly.

I'm not quite ready to publish all of the code (too scared!), but let me know if I can help. It's a very interesting project.

James

PS. Oddly, one of the things I've had the most trouble with (except the bloody autohiding!) is the title bar on docked windows. I've ended up using some really ugly code and a FrameCtrl. How do yours work?

Edit: Exe is for Win32 only, and I've only tested it on XP so I wouldn't be suprised if it crashes horribly on Vista. You have been warned

File Attachments
1) DockTest.exe, downloaded 1124 times

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 22:43:42 GMT
View Forum Message <> Reply to Message

Well It crashes constantly on XP(SP2) too.
Actually, your dockable windows are quite impressing. But the MS style (with all those arrows and I guess you use a transparent window??) is something I wouldn't prefer.

And thank you for the code snippet your workaround seems very simple. I'll try it, I hope it will work.

As for the titlebar, I use a "DragBar" class, which is derived from both Ctrl and CtrlFrame as well. It is a nested class and it differs in implementation for DockWindow and DockBar (Rebar like widget). And almost every event is handled through it. DockWindow is activated only when it is in floating state.

By the way, I implemented the autohide but disabled because of bugs. It's scheduled, and probably will be activated in a week or two

What is the problem with title bar you wrote? Could you be more specific? I could answer any specific questions about DockCtrl implementation. Feel free to ask.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Fri, 28 Sep 2007 23:13:03 GMT
View Forum Message <> Reply to Message

Sorry about the crashing , tonight is the first time I've tried compiling it with MingW (I much prefer MSC but can't get it working here) and it obviously doesn't work very well. I'll repost a version that actually works at some point.

As for the title bars, they work fine. I'm just not happy with the implementation. It's too ugly  Guess I'll have to improve it.

And I agree about the arrows, they are quite unattractive (partly due to my lack of drawing skills), but I think it works quite well as an interface. Implementing other options shouldn't be difficult (like such as a context menu when the drag finishes) but once you introduce tabbing and multi-level docking you need something more complicated than just drag-drop (although this might be preferable for a lot of apps). I'm aiming for the level of complexity that visual studio allows, and since that is the interface it uses it was a good starting point.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Sep 2007 23:29:55 GMT
View Forum Message <> Reply to Message

Well, then we will have both VS and QT/VS style dockable widgets on U++. Actually, I prefer QT syle animations combined with MS style autohide/hide features. So I will take that path.Imho, it's good to have alternative solutions on same platform(as to your dock widget and my dockctrl). This mean we can help each other.
Diversity is a good thing

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Sat, 29 Sep 2007 06:42:06 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 28 September 2007 18:43

As for the titlebar, I use a "DragBar" class, which is derived from both Ctrl and CtrlFrame as well. It is a nested class and it differs in implementation for DockWindow and DockBar (Rebar like widget). And almost every event is handled through it. DockWindow is activated only when it is in floating state.

Beware.

In X11, you cannot use popup windows (I mean, windows without decorations) as main windows.

Consult this for available types:

http://standards.freedesktop.org/wm-spec/wm-spec-1.3.html#id 2507144

(I guess some experimenting will be needed....)

Mirek

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 29 Sep 2007 09:11:51 GMT
View Forum Message <> Reply to Message

luzr wrote on Sat, 29 September 2007 09:42Oblivion wrote on Fri, 28 September 2007 18:43

As for the titlebar, I use a "DragBar" class, which is derived from both Ctrl and CtrlFrame as well. It is a nested class and it differs in implementation for DockWindow and DockBar (Rebar like widget). And almost every event is handled through it. DockWindow is activated only when it is in floating state.

Beware.

In X11, you cannot use popup windows (I mean, windows without decorations) as main windows.

Consult this for available types:

http://standards.freedesktop.org/wm-spec/wm-spec-1.3.html#id 2507144

(I guess some experimenting will be needed....)

Mirek

Hmm, I've read the specifications, what about the _NET_WM_WINDOW_TYPE_TOOLBAR type? Dockwindows are not popups and they are already child windows, so there shouldn't be any problem for them. As for the planned (after I achieve a full featured dockctrl) "DockBar," it is not going to be a "main" window either. Is it not possible to use "child" windows in X11 (I know that terminology differs in X11 due to the conceptualization of it's client-server system, but I'm not a X11 guru)

And I agree. I guess It's time for me to install a linux distro

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 07 Oct 2007 00:52:05 GMT
View Forum Message <> Reply to Message

Ok, here is the new executable.
I've added basic QT-like DnD animation. Finally I've got my hands on two linux distros (Ubuntu-Beryl and Pardus (Turkish)). So I hope that I will release a X11 based exe in coming weeks. (I didn't have the time to install them. (I'll do it probably this week).

Any suggestions, feedbacks, bug reports are welcome.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Sun, 07 Oct 2007 07:43:55 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sat, 06 October 2007 20:52Ok, here is the new executable.
I've added basic QT-like DnD animation. Finally I've got my hands on two linux distros (Ubuntu-Beryl and Pardus (Turkish)). So I hope that I will release a X11 based exe in coming

weeks. (I didn't have the time to install them. (I'll do it probably this week).

Hi! Very impressive! It's almost what I was thinking of (those cool qt animations can be added at the very end). The only missing thing now is tabbed panel and I wonder how tabs should look like. Should they have native look (then you could use tabctrl code to render tabs) or custom one. I think native will be better. Anyway don't stop coding  Results are great!

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mirek on Sun, 07 Oct 2007 08:26:54 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Sat, 06 October 2007 20:52Ok, here is the new executable.

I must have missed something... Where is the new executable?

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Sun, 07 Oct 2007 09:04:55 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 07 October 2007 04:26
I must have missed something... Where is the new executable?

Oblivion always updates first post in this topic.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 07 Oct 2007 09:16:01 GMT
View Forum Message <> Reply to Message

unodgs wrote on Sun, 07 October 2007 12:04luzr wrote on Sun, 07 October 2007 04:26
I must have missed something... Where is the new executable?

Oblivion always updates first post in this topic.

Yes, IMO it's better to keep it that way. I update the executable on the first post every week.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 07 Oct 2007 09:22:44 GMT
View Forum Message <> Reply to Message

unodgs wrote on Sun, 07 October 2007 10:43Oblivion wrote on Sat, 06 October 2007 20:52Ok, here is the new executable.
I've added basic QT-like DnD animation. Finally I've got my hands on two linux distros (Ubuntu-Beryl and Pardus (Turkish)). So I hope that I will release a X11 based exe in coming weeks. (I didn't have the time to install them. (I'll do it probably this week).

Hi! Very impressive! It's almost what I was thinking of (those cool qt animations can be added at the very end). The only missing thing now is tabbed panel and I wonder how tabs should look like. Should they have native look (then you could use tabctrl code to render tabs) or custom one. I think native will be better. Anyway don't stop coding  Results are great!

Well, I prefer native look too. To tell you the truth, it is easy to implement a custom tabctrl. But I think the native U++ style should be preserved (but it is going to be more time consuming). Anyway, I will try to implement a native looking tabctrl (what worries me most is that the native tabctrl has tabs only on top. But we want it to be alignmed dynamically on any side.)

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 07 Oct 2007 16:58:35 GMT
View Forum Message <> Reply to Message

I have added a working TheIDE screenshot with DockCtrl. I couldn't post the modded TheIDE executable, the file limit is 2 MB and I don't know if it will be legal to do so. So if Mirek (I'm calling you Mirek, but hope you dont mind)or any other developers can confirm that it is legal, I will compress and upload a Win32 executable of TheIDE 709-Dev1 asap.

Here is the actual code:

in ide.h

```
// Splitter   pfsplit;
// Splitter   wesplit;

    //============================================================ ============
 // DockCtrl declarations.
 // ---------------------
 // DockCtrl dockctrl (main control)
 //
 // DockWindows are used instead of the splitters:
 // pfsplit   -> packagelistdock, filelistdock;
 // bottom -> bottomdock
    //============================================================ ============

DockCtrl   dockctrl;
DockWindow packagelistdock, filelistdock, bottomdock;
```

in ide.cpp (Ide::SetMain())

```
 //=====================================
 // Actual DockCtrl code...
 //=====================================

if(dockctrl.IsOpen() || dockctrl.IsChild()) dockctrl.Remove();
    Add(dockctrl.Base(this).SetPaneSize(180).Dock(packagelistdoc k.SetLabel(
"Packages").Left().SShow().SetIcon(IdeImg::Package())));
 dockctrl.SetPaneSize(180).Dock(filelistdock.SetLabel("File
List").Left().SShow().SetIcon(IdeImg::Source()));
 dockctrl.SetPaneSize(260).Dock(bottomdock.SetLabel("Bottom
Pane").Bottom().SShow().SetIcon(IdeImg::console()));


 //=====================================
 //
 //=====================================
```

(this is the actual and-will-be syntax of DockCtrl(SetIcon() is optional).

in idewin.cpp 9 lines of code commented out.

```
// editor_bottom.Vert(right_split, bottom);
// editor_bottom.SetPos(8000);

// pfsplit.SetPos(2000);
// pfsplit.Vert(package, filelist);
// wesplit.Vert(pfsplit, editor_bottom);
// wesplit.SetPos(10);
// Add(wesplit);
// packagelistdock.Float();
```

and added:

```
packagelistdock << package.SizePos();
filelistdock  << filelist.SizePos();
bottomdock << bottom.SizePos();
```

And, yes it is working and this easy. Hide bottom pane and hide workspace codes are broken, but the can be fixed by adding

```
dockctrl.ShowxxxPane()
dockctrl.HidexxxPane()
```

codes at appropriate lines.

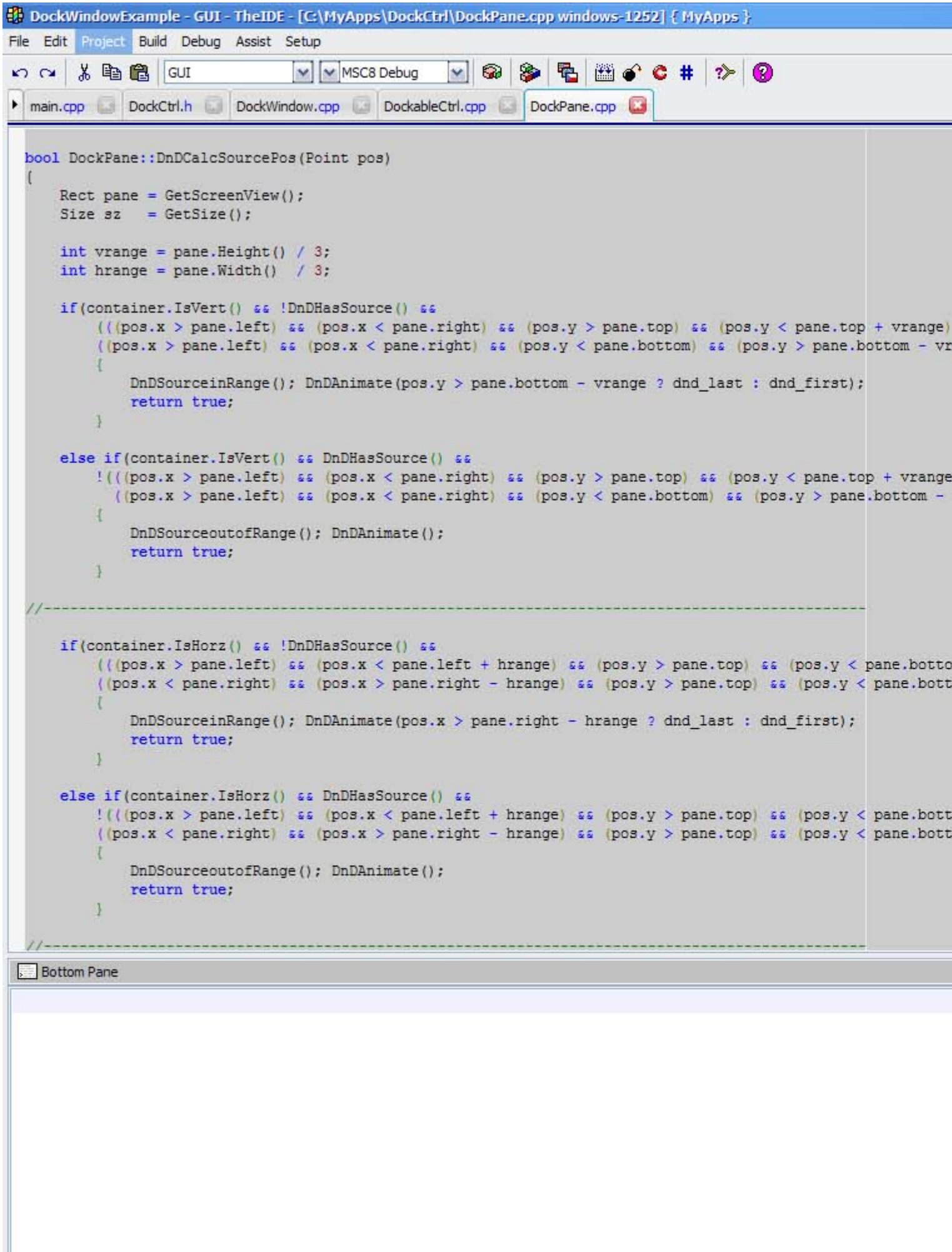
(here xxx is left,right, top, bottom (eg. ShowLeftPane())

(probably there are more difficulties that I will encounter, but it's neither TheIDE's nor DockCtrl's fault).

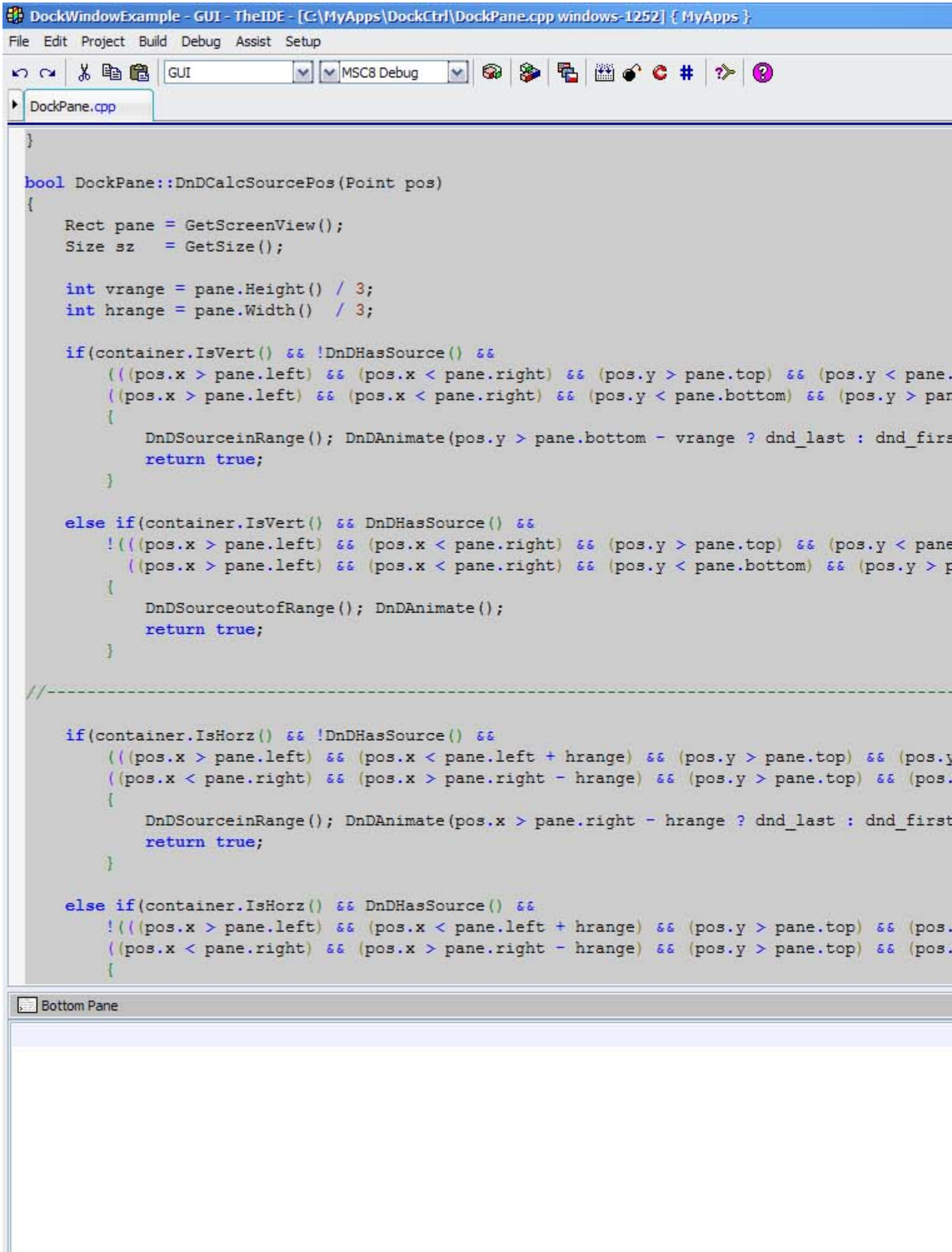
Regards

## File Attachments

1) DockCtrlWrapped TheIDE.jpg, downloaded 1589 times


DockWindowExample - GUI - TheIDE - [C:\MyApps\DockCtrl\DockPane.cpp windows-1252] { MyApps }
File Edit Project Build Debug Assist Setup
GUI
MSC8 Debug
main.cpp
DockCtrl.h
DockWindow.cpp
DockableCtrl.cpp
DockPane.cpp
bool DockPane::DnDCalcSourcePos(Point pos)
{
Rect pane = GetScreenView();
Size sz = GetSize();
int vrange = pane.Height() / 3;
int hrange = pane.Width() / 3;
if(container.IsVert() && !DnDHasSource() &&
(((pos.x > pane.left) && (pos.x < pane.right) && (pos.y > pane.top) && (pos.y < pane.top + vrange)
((pos.x > pane.left) && (pos.x < pane.right) && (pos.y < pane.bottom) && (pos.y > pane.bottom - vr
{
DnDSourceinRange(); DnDAnimate(pos.y > pane.bottom - vrange ? dnd_last : dnd_first);
return true;
}
else if(container.IsVert() && DnDHasSource() &&
!(((pos.x > pane.left) && (pos.x < pane.right) && (pos.y > pane.top) && (pos.y < pane.top + vrange
((pos.x > pane.left) && (pos.x < pane.right) && (pos.y < pane.bottom) && (pos.y > pane.bottom -
{
DnDSourceoutofRange(); DnDAnimate();
return true;
}
//
if(container.IsHorz() && !DnDHasSource() &&
(((pos.x > pane.left) && (pos.x < pane.left + hrange) && (pos.y > pane.top) && (pos.y < pane.botto
((pos.x < pane.right) && (pos.x > pane.right - hrange) && (pos.y > pane.top) && (pos.y < pane.bott
{
DnDSourceinRange(); DnDAnimate(pos.x > pane.right - hrange ? dnd_last : dnd_first);
return true;
}
else if(container.IsHorz() && DnDHasSource() &&
!(((pos.x > pane.left) && (pos.x < pane.left + hrange) && (pos.y > pane.top) && (pos.y < pane.bott
((pos.x < pane.right) && (pos.x > pane.right - hrange) && (pos.y > pane.top) && (pos.y < pane.bott
{
DnDSourceoutofRange(); DnDAnimate();
return true;
}
//
Bottom Pane

2) DockCtrlWrapped TheIDE 2.jpg, downloaded 1590 times


DockWindowExample - GUI - TheIDE - [C:\MyApps\DockCtrl\DockPane.cpp windows-1252] { MyApps }
File Edit Project Build Debug Assist Setup
GUI
MSC8 Debug
DockPane.cpp
}
bool DockPane::DnDCalcSourcePos(Point pos)
{
Rect pane = GetScreenView();
Size sz = GetSize();
int vrange = pane.Height() / 3;
int hrange = pane.Width() / 3;
if(container.IsVert() && !DnDHasSource() &&
(((pos.x > pane.left) && (pos.x < pane.right) && (pos.y > pane.top) && (pos.y < pane.
((pos.x > pane.left) && (pos.x < pane.right) && (pos.y < pane.bottom) && (pos.y > pan
{
DnDSourceinRange(); DnDAnimate(pos.y > pane.bottom - vrange ? dnd_last : dnd_firs
return true;
}
else if(container.IsVert() && DnDHasSource() &&
!(((pos.x > pane.left) && (pos.x < pane.right) && (pos.y > pane.top) && (pos.y < pane
((pos.x > pane.left) && (pos.x < pane.right) && (pos.y < pane.bottom) && (pos.y > p
{
DnDSourceoutofRange(); DnDAnimate();
return true;
}
//----------
if(container.IsHorz() && !DnDHasSource() &&
(((pos.x > pane.left) && (pos.x < pane.left + hrange) && (pos.y > pane.top) && (pos.y
((pos.x < pane.right) && (pos.x > pane.right - hrange) && (pos.y > pane.top) && (pos.
{
DnDSourceinRange(); DnDAnimate(pos.x > pane.right - hrange ? dnd_last : dnd_first
return true;
}
else if(container.IsHorz() && DnDHasSource() &&
!(((pos.x > pane.left) && (pos.x < pane.left + hrange) && (pos.y > pane.top) && (pos.
((pos.x < pane.right) && (pos.x > pane.right - hrange) && (pos.y > pane.top) && (pos.
{
Bottom Pane

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 13 Oct 2007 12:13:50 GMT
View Forum Message <> Reply to Message

Hi,
This week I've managed to add default and custom chameleon/skin support for DockWindow. From now on, it is possible to skin the dragbar and dragbar buttons. By default, there are 3 skins: classic, enhanced and modern. By default, DockWindow class uses modern skin. But it can be changed on-the-fly through Control Panel. I've also added a custom skin support, so that programmer can add his/her custom skin to the skin list. (Hope you'll like it). For other changes and example executable, please look at the change log. It's in the first post of this topic.

Any feedback, suggestions, hints and constructive criticism are welcome.

Regards.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Sat, 13 Oct 2007 15:10:19 GMT
View Forum Message <> Reply to Message

New buttons are quite pretty, but IMO too much rounded. I found you fixed inserting window "in the middle". I mean in the previous version if there were 3 or more docked windows in a row it wasn't possible to insert dragged window at 2nd or 3rd position - only at the beginning or the end. Now I can  but if I drag the widnow and "docking system" shows blue rectangle and in the meantime mouse goes over a splitter bar - blue rectangle disappear and appears again as soon as mouse is not longer over the splitter. This causes unpleasant effect some kind of flickering. If you could eliminate that please do it .
The second problem is right docked window. It's not possible to get it and put it in the same position (to make it fills the whole right area (from top to the bottom of the window)).
If something of it it's unclear to you please ask for further explanations.
All in all we're very close to achieve basic functionality. Great progress!

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Sat, 13 Oct 2007 15:19:09 GMT
View Forum Message <> Reply to Message

One more bug. Run your example. Now grab the right panel and move it to the top right corner of the window (do not release the left mouse button). You should see blue rectangle in this corner. Now move the mouse down. You should observe a lot of flickering.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 13 Oct 2007 16:07:19 GMT

Quote:New buttons are quite pretty, but IMO too much rounded.

Well I'm not very good at drawing and graphics. If someone would provide me with a better image/icon set, I will gladly change the so-called "skins."

Quote:if I drag the widnow and "docking system" shows blue rectangle and in the meantime mouse goes over a splitter bar - blue rectangle disappear and appears again as soon as mouse is not longer over the splitter. This causes unpleasant effect some kind of flickering. If you could eliminate that please do it Smile.

Yes I will eliminate the flickering problem in this week. It's very annoying indeed.

Quote: The second problem is right docked window. It's not possible to get it and put it in the same position (to make it fills the whole right area (from top to the bottom of the window)).

Oops, my fault. I was so busy with the dock positioning and chameleon that I forgot to implement the ShowPane() method for empty panes (It gets locked when it's empty and there is no dnd-source). I'll fix it asap.

And, Thank you for your bug reports and feedback.

regards

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 27 Oct 2007 10:01:08 GMT


Ok, here we are again. Sorry for the long delay. I was very busy in last week. So you will only find a number of bugfixes in the current example (0.49.9.9a). The good news I think we are much more closer to the first publication of the source code. I've finally installed the two Linux derivatives (Ubuntu (with beryl) and Pardus) and I will probably compile the example on them this, or next week. I've implemented a qt-like animation but did not have the time to test it, so it will stay disabled for a little while. As for the skins, Classic skin is reverted to the standard system color, so I think it now has more native-look. I've also added to one of the dockwindows a simple "layout." (to show that they are fully in U++). The Tabbed dockwindows are finally in progress too...
Ps: I am well aware that Drag-Drop animations and sizing are not very good. They will be improved, asap)

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 07 Nov 2007 19:18:53 GMT

View Forum Message <> Reply to Message

Hi. Sorry for delayed answer. 0.49.9.9a (how do you know what particular number is for ) is great. I mean all my bugs are in fact gone. I like the new blue background when window is detached. I'm waiting for a new version!

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 10 Nov 2007 23:16:01 GMT
View Forum Message <> Reply to Message

Hi, I am sorry for the delay (both for the answer and for the new exe). But, unfortunately, I am still too busy with my school and my job, so I am going to post a new executable (hopefully with the linux/X11 exe this time) next week. As for the version numbers, I am going to sync it with the U++ and use U++ versioning style (e.g. 711a) when I release the source code

p.s.: Don't be mistaken by the slow down. Progress is only temporarily slowed down. I will not drop the project (actually, it is in progress at the background).

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Mon, 12 Nov 2007 14:40:35 GMT
View Forum Message <> Reply to Message

How are you doing on the X11 port? I've been looking at the problem myself and it looks like X11 doesn't supply the necessary events to implement window dragging as cleanly as on Windows. The only available event is the equivalent of WM_MOVE, and you can't get the status of the mouse buttons during the drag without calling XQueryPointer, which should probably be avoided.

I've seen it suggested that Qt requires Window Manager specific code for this (I haven't checked the source myself) which, if true, would mean that a 'pure' X11 implementation may be impossible. Additionally, changes to TopWindow would be necessary to support the _NET_WM_WINDOW_TYPE_DOCK style.

In case it helps, here is the code that starts window dragging after an undocking:

```
Atom xwndDrag = XAtom("_NET_WM_MOVERESIZE");
XEvent e;
Zero(e);
e.xclient.type = ClientMessage;
e.xclient.message_type = xwndDrag;
e.xclient.window = wnd.GetWindow();
e.xclient.format = 32;
e.xclient.display = Xdisplay;
e.xclient.send_event = XTrue;
e.xclient.data.l[0] = p.x;
e.xclient.data.l[1] = p.y;
e.xclient.data.l[2] = 8;
e.xclient.data.l[3] = 1;
```

```
e.xclient.data.l[4] = 0;

XUngrabPointer( Xdisplay, CurrentTime );
XSendEvent(Xdisplay, RootWindow(Xdisplay, Xscreenno), XFalse, SubstructureNotifyMask, &e);
XFlush(Xdisplay); It is the same as doing:
SendMessage(wnd.GetHWND(), WM_NCLBUTTONDOWN, 2, MAKELONG(p.x, p.y));with
```
Win32.

If you find/have found a solution to the window dragging issue I'd love to hear about it

James

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Mon, 12 Nov 2007 22:12:50 GMT
View Forum Message <> Reply to Message

Thank you very much mrjt. To be honest, since the only platform specific implementation is about window dragging and moving, this issue is the cause of the delay of the X11 executable. I'm trying to find a workaround for it. After I achieve at something concrete, I would be glad to share my experience with you

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Tue, 13 Nov 2007 15:14:33 GMT
View Forum Message <> Reply to Message

Thanks. The reason I'm interested is that I'm still working on a version of this myself (for fun as much as anything else). I've attached a demo, which is pretty much feature complete barring the X11 port. I've decided that your drag-drop approach is better than VS style popups, but it's not quite as pretty as yours

Known issues:
Autohide popup alignment is incorrect on right/bottom bars (I have no idea why this is, it varies depending on the window).
Sizing of docked windows isn't intelligent enough yet.
Tabs are quite ugly (I'm waiting for the new Chameleonized QuickTabs so I can steal the drawing code ).
Still not tested on Vista, so please let me know if there are problems.

James

File Attachments
1) DockTest.zip, downloaded 459 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 13 Nov 2007 20:51:12 GMT
View Forum Message <> Reply to Message

To be honest, your widget is far better than mine (at least, in functionality ) And I'm really impressed by the progress. I think i will need your help (If you dont mind) for the autohide feature. Mine is not very good Which ctrl did you use to create that sliding pane?

ps: There is a bug in your docktest.exe which leads to crash on Xp and Vista (basic):

When I try to undock (float) a tabbed window via the dropdown menu ("floating"), AND when it asks: "There are multiple tabs open, which would you like to float", if I select the "All other tabs" option, the example crashes immediately.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 14 Nov 2007 08:10:35 GMT
View Forum Message <> Reply to Message

I see we have two competiting solutions Great I think you should work together instead of developing the same thing separately. Maybe we should add DockCtrl to uvs tree so you could easily cooperate.
I also think that Oblivion's docking system is prettier and more intuitive. I think following qt docking system is the best way. Qt guys did great job there. As for tabs. I could modify quick tabs to be more universal widget that could draw tabs in any direction and support drag and drop in any direction too. Of course I don't have any objections to coping paint routines to your code if you want to develeop the whole system on your own now.
Anyway both of you are doing a great job. Just don't stop This will be very important part of upp.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Wed, 14 Nov 2007 10:46:35 GMT
View Forum Message <> Reply to Message

OblivionAnd I'm really impressed by the progress.Thank you, I guess I've had more time do spend ont it than you have.

Oblivionps: There is a bug in your docktest.exe which leads to crash on Xp and Vista (basic):Thanks, I've fixed it and re-uploaded.
OblivionWhich ctrl did you use to create that sliding pane?It's a popup that gets resized to do the animation. I may have to change this as I'm not sure it will work on Linux.
unodgsI also think that Oblivion's docking system is prettier and more intuitive.It's definitely prettier, but as everything is drawn with chameleon this is easy to change. I'm not sure what you mean by more intuitive because if you enable simple docking and disable tabbing they work

almost identically.

The complex docking (it works something like a tree of splitter ctrls) is actually the main reason I was developing this myself - I didn't think anyone else would and I wanted it
unodgsI think following qt docking system is the best way. Qt guys did great job there.I've only seen the examples, but I disagree. I think they are fairly ugly, missing features and even contain some small bugs. We can do much better.

The only things currently missing to match the Qt feature set is the animation and some configuration options. After that I have a list of additional features that I want.
unodgsI could modify quick tabs to be more universal widget that could draw tabs in any direction and support drag and drop in any direction too.
This was partly my intention as I think Upp would be improved by having a general QuickTabs frame, which I could then use. But I think you would do a much better job at this than me!
unodgsI think you should work together instead of developing the same thing separately.
I agree, but since niether of us has posted the source this has been impossible. I want to do some tidying/restructuring first (it's complicated code and gets a bit messy in places) and I'm away this weekend, but I'll hopefully post it next week.

The next problem is the X11 port. I have a bad feeling about this

James

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 14 Nov 2007 12:15:31 GMT
View Forum Message <> Reply to Message

Quote:I'm not sure what you mean by more intuitive because if you enable simple docking and disable tabbing they work almost identically.
Yes, you're right. Didn't notice that switch  I toyed a little bit more with your app and found that in not simple docking mode you allow to divide window "in four ways" (I hope you know what I mean ). And I know why it looked strange at first sight to me. If for example you drag a window to the bottom part of a main window and there is another widnow the height of these 2 widnows should be summed up not divided by 2.
Anyway I think we could have this 2 docking systems built-in (with some switches like calculating this height/width mode for example)
Quote:I've only seen the examples, but I disagree. I think they are fairly ugly, missing features and even contain some small bugs. We can do much better.
The only things currently missing to match the Qt feature set is the animation and some configuration options. After that I have a list of additional features that I want.
Try mainwidnows example. And of course we could be better. I'll modify my previous statement. We should at least achieve qt level  And in fact only those animations are not implemented yet.
Quote:This was partly my intention as I think Upp would be improved by having a general QuickTabs frame, which I could then use. But I think you would do a much better job at this than me!
Ok, just tell me (you and oblivion) what interface you need (eg what callbacks should be there and when they should be called).

Quote:I agree, but since niether of us has posted the source this has been impossible. I want to do some tidying/restructuring first (it's complicated code and gets a bit messy in places) and I'm away this weekend, but I'll hopefully post it next week.

So post the code right now  (you can sand it to me by email if you don't want to make it public yet) I'll move it to our repository tree and give you instructions how to use it.
Quote:
The next problem is the X11 port. I have a bad feeling about this
I guess so. But let's have it done in win32 first at least.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Wed, 14 Nov 2007 15:47:38 GMT
View Forum Message <> Reply to Message

I guess I may as well post it. Just don't look to closely at the code

unodgsIf for example you drag a window to the bottom part of a main window and there is another widnow the height of these 2 widnows should be summed up not divided by 2.
This is what I meant by sizing needs more intelligence. It works reasonably well in simple mode (though it still needs improvement), but becomes difficult to work out what's going on in more complex cases and I haven't sorted it out yet. One reason for some restructuring is to make this easier.

unodgsOk, just tell me (you and oblivion) what interface you need (eg what callbacks should be there and when they should be called).
My requirements aren't great. Something like:

```
Callback  WhenSelect;
Callback  WhenHighlight;
Callback  WhenDrag;
Callback  WhenContext;
Callback       WhenTabClose;

QuickTabs &AddTab(Value &key, Value &list, Display *display = NULL);

QuickTabs &NoSelection(bool s = true);
QuickTabs &NoHighlight(bool h = true);
// Plus usual get/set stuff
```

Plus some caveats:
Drag-drop needs to be overridable, because I need dragged tabs to turn into windows, not start a DnD op.
Height (width if vertical) must be scaleable in some way.
I think everything specialised I need could then be added by an inheriting class (Autohide, storing Ctrls etc.). You can see my TabBar class for exactly what I use, I hadn't really got as far as generalising it.

James

File Attachments
1) Docking.zip, downloaded 450 times

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 14 Nov 2007 16:08:26 GMT
View Forum Message <> Reply to Message

Quote:So post the code right now Smile (you can sand it to me by email if you don't want to make it public yet) I'll move it to our repository tree and give you instructions how to use it.

Ok then, so here's the deal

I will gladly send you (and/or mrjt) the source code of my "dockctrl" implementation. And I will send you the 049.9.9a (package used in the last executable I've uploaded )because 0.50 (with tabs and autohide) is in progress, and the code is not stable yet . I removed all the buggy autohide system from the source). I'm not posting here; not because I consider it to be a "top secret" government project, but because I think that the source code is not mature enough, and as mrjt stated, it has to be refactored/restructured (and it will be). But please don't expect a miracle; the code is, imho, fairly simple for a such project (that's a reason why I'm not confident to publish it yet, you may find it it almost "silly" when you examine it).

But please give me some feed back, I want to know your idea about the code. Only then, if you find it reasonably good, I will be willing to see it on the uvs of U++.

And one more thing. I am considering to convert the names of the main classes.
Is it not better to make the main ctrl (dockctrl), a window rather than a ctrl. I mean, currently the docs are called DockWindow and the main ctrl is called DockCtrl. Would it not be better if the main ctrl has the name DockWindow? And would it not be better if an application which wants to use docking system, should derive its main window from the DockWindow rather than a TopWindow ? Imho, this is a better approach. What would you think about it?

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Wed, 14 Nov 2007 16:54:53 GMT
View Forum Message <> Reply to Message

A window would probably be better, though the way I've done it you can have either (have a look at bottom of DockBase.h). The only problem here is that any MenuBars/StatusBars added to the window need to be added before the SplitterFrames.

Subject: Re: DockCtrl (A dockable window widget for U++)

Posted by Oblivion on Tue, 15 Jan 2008 01:03:23 GMT
View Forum Message <> Reply to Message

At last, here we are again

I know It's been a long while since I last updated the DockCtrl. I was toooo busy. DockCtrl is under heavy development in the background. But I feel obliged to post something just to show that the project is not stopped. Since I am now free to focus on this project, I will update it regularly (weekly) again.

I've just uploaded a "preview" of ver 0.50 (this will be the last version with all the dottes, and numbers. I am going to syncronise it with the Upp versioning system).

From now on, dockctrl will have a sophisticated tabbing mechanism. Thanx to Unodgs for "QuickTabs" . I modified it and used it in dockctrl (From now on, it is also a generic TabCtrl interface called TabInterface -- which can be used and customised in any project. So, maybe I will upload this new "widget" seperately after I polish it. The results are good, I think.

- This preview has the tabs only under tabdocks. The final version of 0.50 will have tabs on left/right sides too (its in progress).

- Also, some sort of "recursive tabbing" is introduced. What I mean is, tabbed docks can be tabbed as groups under other tabdocks (It looks like "cascaded" tabs) and this can be done by drag-n-drop with mouse (they can also be dragged as groups). I think this can save a lot of space in Photoshop-like big application.
Since it uses a modification of QuickTabs, it has the "scroll" ability too. You can scroll tabs (with scroll bar or with middle mouse button), when they are not visible, as it is in theIDE.
Try and see it...

- Every tab can have icons (one for custom icon, and one for close icon) and has a a traditional dockctrl command menu which can be invoked by right clicking on tab. And, "left dragging" will tear the tab.

Please keep in mind that this is only a "preview" version. I know there are hell lot of bugs and somethings are broken (e.g. ResetLayout() and autosizing() are broken needs reworking)

Also, the source code is very messy, and scattered at the moment, and will go under heavy restructuring.So I decided not to commit to SVN repository until I clean the mess.

Bug reports, suggestions, constructive criticisms are always welcome.

- No complete X11 support yet.
- No AutoHide.

Ps: as always is, you can find the example preview exe is in the first post of this topic.

Regards.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Tue, 15 Jan 2008 07:59:37 GMT
View Forum Message <> Reply to Message

Quote:Since it uses a modification of QuickTabs, it has the "scroll" ability too. You can scroll tabs (with scroll bar or with middle mouse button), when they are not visible, as it is in theIDE.
Try and see it..
I started to work on quicktabs some time ago. They would support every paint direction (so you could paint tabs of hidden panels) and autohide of srcollbar. I'll integrate it (with your support of course) with your code.
I'll try to finish it very soon. I was (still am actually..) very busy with my comercial project and that's why I was unable to do it earlier.
Thanks for new release.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by forlano on Tue, 15 Jan 2008 08:40:07 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Tue, 15 January 2008 02:03
Bug reports, suggestions, constructive criticisms are always welcome.

I tried the executable and I found the ctrl excellent. The photoshop like tabs work very nice. I would like to use this features as soon as possible.

Luigi

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Tue, 15 Jan 2008 13:34:37 GMT
View Forum Message <> Reply to Message

unodgs wrote on Tue, 15 January 2008 07:59I started to work on quicktabs some time ago. They would support every paint direction (so you could paint tabs of hidden panels) and autohide of srcollbar.
Please finish this! I thought you'd givien up so I have a version that is almost complete (I'm having some problems with the close buttons when left aligned) but I expect yours will be much cleaner. At the moment I'm still stuck with the ugly old style QuickTabs

I've also updated my version of the Docking stuff, but I thought I'd post in Bazaar and let Oblivion have his thread back

Ps. Nice idea with the nested tabs.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Tue, 15 Jan 2008 19:58:27 GMT
View Forum Message <> Reply to Message

Don't worry. I'll finish it. It's better to have one tab system. Of course I'll take a look at your and Oblivion's code to see what you need and how you use your current solutions.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 17 Feb 2008 04:00:43 GMT
View Forum Message <> Reply to Message

Hi, finally I have updated the DockCtrl to ver 0.50. Now DockCtrl has all the basic functionality for Win32 platform.

Also, from now on, I syncronize the releases with monthly U++ releases, so it is actually DEV802b.1. Namely, it is in Beta stage So, I am officially supporting the ctrl now and in the future.

This is a major update. You can find the details of the update and the example win32 exe in the first post of this topic. Also, I finally updated the SVN and added the self-explanatory source code of the example exe.

From this version on, DockCtrl has full AutoHide and Tabbing support. The "nested tabbing" is made optional and can be switched on, on-the-fly. Also, tabs has "automatic alignment" feature. by default tabs are bottom-aligned. But this can be changed on-the-fly by the user. In automatic alignment mode, Every tabwindow has it's tabs on the opposite side of it's alignment (e.g, LEFT/right, TOP/bottom and vice versa)

Chameleon (skins) method was broken, now fixed. Also, DockCtrl now has a unified chameleon structure for easy customizing (not complete yet).

I am planning to add internationalization support and (actually, it does have support; all I have to do is to add a *.t file -- I'm a bit lazy nowadays ) write documentation, add X11 support.
========== ===
The Easter EGG:
One of the most interesting "feature" of DockCtrl is that now it is Ctrl dependent, therefore it can be "nested". This means, theoretically, you could have infinite number of independent dockctrl framework (yes, not only dockable windows, but also the framework itself in a single application window! Namely, even DockCtrl could be "nested" too! For example, you could even have child DockCtrl frameworks in Dock widgets(!) or other standard U++ or CtrlFrame derived classed! I will write an example code to demonstrate this "feature" (All you have to do is to pass the target Ctrl reference to SetLayout() method. Thats all.

========== ===

As always is, there should be some bugs (I didn't encounter any yet). I need feedback, so please feel free to report bugs you encounter while playing with the exe. I'm also going rename some classes and methods (I know they are not well named).

Also, code needs refactoring.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Mon, 18 Feb 2008 02:22:38 GMT
View Forum Message <> Reply to Message

Quote:
I am planning to add internationalization support and (actually, it does have support; all I have to do is to add a *.t file -- I'm a bit lazy nowadays Embarassed )

Well, It's time to rise and shine. I have added the *.t translation file to the SVN tree too.

From now on, DockCtrl has internationalization support. At the moment, only English and Turkish.

I need help for translations (there are only 15-20 lines to translate, so it shouldn't be time consuming)...

Here I upload the translation file. Any help will be appreciated.
Thank you.

Regards

File Attachments
1) DockCtrl.t, downloaded 486 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 20 Feb 2008 00:48:44 GMT
View Forum Message <> Reply to Message

I've updated the DockCtrl SVN package to version DEV-802b.2 (0.51) and the Example exe code (example exe is localized. But unfortuantely, at this point only Turkish and English languages are fully supported.I'm also going to add French, if someone doen't -- but I prefer a native French speaker ). Also, I've added changelog and todo list to the source. As usual, you can find the changelog and new Example exe on the first post of this topic. It has all the main features

enabled. Still, you may encounter some problems with the "nested tabbing", It now has the top priority on my todo list and I'm still working on it. Also I need some feedback.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Wed, 20 Feb 2008 12:23:16 GMT
View Forum Message <> Reply to Message

Hi,

the attached archive contains modified DockCtrl package fixing some issues with MingW 3.4.2 under WinXP (mostly call-by-reference issues) and also providing german translations.

Matthias

File Attachments
1) DockCtrl.zip, downloaded 454 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 20 Feb 2008 12:42:51 GMT
View Forum Message <> Reply to Message

Quote:
Hi,

the attached archive contains modified DockCtrl package fixing some issues with MingW 3.4.2 under WinXP (mostly call-by-reference issues) and also providing german translations.

Matthias

Thank you very much. I will add the changes to the current package immediately.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 20 Feb 2008 13:37:49 GMT
View Forum Message <> Reply to Message

I have fixed (I think so) all tabs painting issues as well as detecting close button position. Fixes are already in svn.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 20 Feb 2008 13:56:28 GMT
View Forum Message <> Reply to Message

Quote:
I have fixed (I think so) all tabs painting issues as well as detecting close button position. Fixes are already in svn.

Thanks Daniel. I will merge it with the current version.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Wed, 20 Feb 2008 14:06:18 GMT
View Forum Message <> Reply to Message

There is a template parameter missing in QuickTabs.cpp line 87:
FrameCtrl::FramePaint(w, r);
should be:
FrameCtrl<Ctrl>::FramePaint(w, r);

Matthias

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 20 Feb 2008 14:08:23 GMT
View Forum Message <> Reply to Message

Thanks!. That's because I mostly use MSVC.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 20 Feb 2008 15:25:05 GMT
View Forum Message <> Reply to Message

Quote:
I have fixed (I think so) all tabs painting issues as well as detecting close button position. Fixes are already in svn.

I've applied the patches and added the German translations to the source. By the way, Daniel could you please update the changelog when you fix or add stg. It's easy to track the changes/fixes there.

Also, if you have spare time (It won't take more than 10 minutes) could you please add Polish translation to the source (AFAIK you're Polish)? I Would be grateful.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 20 Feb 2008 18:00:04 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Wed, 20 February 2008 10:25
I've applied the patches and added the German translations to the source. By the way, Daniel could you please update the changelog when you fix or add stg. It's easy to track the changes/fixes there.

Naturaly, no problem. I love writing changelogs myself too
Quote:
Also, if you have spare time (It won't take more than 10 minutes) could you please add Polish translation to the source (AFAIK you're Polish)? I Would be grateful.
Sure.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Thu, 21 Feb 2008 01:19:48 GMT
View Forum Message <> Reply to Message

Finally I find a way to draw dragbar skins with native/system chameleon values. From now on DockCtrl's skins adapt themselves to the System's visual styles. And I've updated both SVN and the example exe in the first topic with some cool visual samples.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by cbpporter on Thu, 21 Feb 2008 09:05:55 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Thu, 21 February 2008 03:19Finally I find a way to draw dragbar skins with native/system chameleon values. From now on DockCtrl's skins adapt themselves to the System's visual styles. And I've updated both SVN and the example exe in the first topic with some cool visual samples.

This is getting better and better. A couple of more such minor improvements and I"ll give it a try in some production code, to see how it behaves in the long run.

There's one small issue, I don't know if you noticed yet. Mark a panel as autohide and move your mouse over the border tab so that the panel appears with animation. All goo up to here. But if you quickly move the whole window, the poped-up panel remains in place until it disappears.

Subject: Re: DockCtrl (A dockable window widget for U++)

Posted by Oblivion on Thu, 21 Feb 2008 09:35:42 GMT
View Forum Message <> Reply to Message

Quote:

There's one small issue, I don't know if you noticed yet. Mark a panel as autohide and move your mouse over the border tab so that the panel appears with animation. All goo up to here. But if you quickly move the whole window, the poped-up panel remains in place until it disappears.

No I didn't noticed that. Thank you. I will fix it asap.

Quote:
This is getting better and better. A couple of more such minor improvements and I"ll give it a try in some production code, to see how it behaves in the long run.

Well, as a matter of fact, I 'm currentyl using DockCtrtl in TheIDE. And apart from some small bugs in normal mode (no nested tabbing - it still needs to be refined), it seems to work fine. But again, DockCtrl is still under heavy development. It seems that the famous "chicken-egg" problem is a traditional situation of the U++ apps

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Thu, 21 Feb 2008 11:15:18 GMT
View Forum Message <> Reply to Message

And window drag and drop should not work if I start dragging the tab. The tab itself should be drageed instead in this particular case. If this and window animation (this could be stolen from mrjt )will be done it's all, right? How about linux?

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Thu, 21 Feb 2008 12:23:01 GMT
View Forum Message <> Reply to Message

You'll have a difficult job stealing my animation code! It's both horifically complicated (due to the tree-like nesting and size-hints) and implemented using a different ctrl. It's not too hard though, IMO the only real problem will be persuading the Splitter to proportionally resize it's children instead of just the animated one.

I'm going to steal your theming code though Nice work!

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Thu, 21 Feb 2008 12:38:43 GMT
View Forum Message <> Reply to Message

unodgs wrote on Thu, 21 February 2008 11:15And window drag and drop should not work if I start dragging the tab.
I'm not sure about this. IMO in this situation the user is much more likely to want to move the window than re-arrange the tabs. It should be possible to drag the tabs though (works while holding CTRL key in my version).

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Thu, 21 Feb 2008 13:18:05 GMT
View Forum Message <> Reply to Message

Quote:
You'll have a difficult job stealing my animation code! It's both horifically complicated (due to the tree-like nesting and size-hints) and implemented using a different ctrl. It's not too hard though, IMO the only real problem will be persuading the Splitter to proportionally resize it's children instead of just the animated one.

I'm going to steal your theming code though Smile Nice work!

Well, I won't steal too much  I've already implemented a animation system (not in the public version). It's hard but not that hard. I Since I use the splitter as the baseclass of panesplitter, I overrided the layout() method -- i think it's good place to start. But I hate that "10000" bla bal proportion thing

Also, if want to see the "magic" chameleon code just look inside the DockWindow::DraBar::Paint() and DockCtrlChStyle::StyleDefault()you won'belive how easy it is implemented (how fool was I not see it before  )

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Thu, 21 Feb 2008 13:27:37 GMT
View Forum Message <> Reply to Message

Quote:
And window drag and drop should not work if I start dragging the tab. The tab itself should be drageed instead in this particular case. If this and window animation (this could be stolen from mrjt

Wink )will be done it's all, right? How about linux?

Well, I disagree. I intentionally removed the tab dragging code. Of course I can change that, but wouldn't it be better if the tabs could be rearranged using "middle button" drag (It'a piece of cake to implement this)? As for the linux thing, that what I'm going to steal from mrjt  I've already examined his X11 code and I have to say that He's done a great job there. It's also very easy to adopt by DockCtrl.

I have a lot of spare time this month, so probably I'll finish the code at the end of March (hopefully)

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Thu, 21 Feb 2008 14:04:57 GMT
View Forum Message <> Reply to Message

mrjt wrote on Thu, 21 February 2008 07:38
I'm not sure about this. IMO in this situation the user is much more likely to want to move the window than re-arrange the tabs. It should be possible to drag the tabs though (works while holding CTRL key in my version).

IMO when you start to drag tab you WANT to rearange tab position, if you drag window top bar you WANT to drag just window. This is how qt docking system works. And Visual Studio too. You can also think of moving tab like about moving window because window is assigned to the tab.
Anyway. To satisfy all of us I propose:
If user start dragging a tab and mouse cursor is in tabs area you dragging the tab, if mouse cursor exceedes tabs area (+5-10 pixel border) user start dragging the window assigned to this tab.
If I hold ctrl tab is dragged no matter what. What do you think?
I only wonder if it's possible to implement with current upp's D&D api.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Thu, 21 Feb 2008 14:43:16 GMT
View Forum Message <> Reply to Message

Quote:This is how qt docking system works. And Visual Studio too.
The version of VS I have installed (VS Express 2005) works like this (I just checked):
Drag title bar: Detached/move whole window including all tabs
Drag a tab: Detach single tab
Drag a tab with SHIFT key: rearrange tabs

The Qt demo I have (version 4.x, but not the most recent):
Drag title bar: Detaches a single tab
Drag a tab: Does nothing
Drag a tab with SHIFT key: Does nothing (also CTRL does nothing)
Also, Qt doesn't seem to have the ability to detach a whole collection of tabs (hence why dragging

the handle undocks only one tab). I don't have much experience beyond looking at some of the source with the Qt system though, so please correct me if I'm wrong.

I think we have it right personally.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Thu, 21 Feb 2008 17:06:59 GMT
View Forum Message <> Reply to Message

You're right. I checked qt example. I don't know why I saw dragable tabs  I'm sure in VS 2003 I could drag tabs. So let's stay with dragging with Ctrl key pressed but we could add a switch to change this behaviour of course.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Thu, 21 Feb 2008 18:27:32 GMT
View Forum Message <> Reply to Message

If you want to have a play with how I think the tabs should work I've attached my latest stable build. I've just finished implementing theming (Thanks Oblivion!), and nested tabs. Hold down CTRL while dragging a tab collection to add it as nested (there is a switch to change the default behaviour but I haven't added to the example since I'll have to make a better on soon anyway).

I've also enabled tab grouping, with grayed-out tabs showing inactive (I find quicktabs just hiding them to be a bit confusing).

Hold CTRL to drag tabs. There may be some bugs.

James

File Attachments
1) DockTest.zip, downloaded 484 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Thu, 21 Feb 2008 19:27:49 GMT
View Forum Message <> Reply to Message

Quote:
IMO when you start to drag tab you WANT to rearange tab position, if you drag window top bar you WANT to drag just window. This is how qt docking system works. And Visual Studio too. You can also think of moving tab like about moving window because window is assigned to the tab. Anyway. To satisfy all of us I propose:
If user start dragging a tab and mouse cursor is in tabs area you dragging the tab, if mouse cursor

exceedes tabs area (+5-10 pixel border) user start dragging the window assigned to this tab.
If I hold ctrl tab is dragged no matter what. What do you think?
I only wonder if it's possible to implement with current upp's D&D api.

Why are you so obsessed with the draggable tabs (just kidding) . Well if it suits you, I will add it to my todo list on with high priority. I will add a switch for this. DockCtrl framework is (you may have noticed already) very flexible and it's easy to implement this

Quote:
You're right. I checked qt example. I don't know why I saw dragable tabs Rolling Eyes I'm sure in VS 2003 I could drag tabs. So let's stay with dragging with Ctrl key pressed but we could add a switch to change this behaviour of course.

What about implementing Ctrl OR Middle Mouse Button Drag ?

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Thu, 21 Feb 2008 19:54:26 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Thu, 21 February 2008 14:27
Why are you so obsessed with the draggable tabs (just kidding) . Well if it suits you, I will add it to my todo list on with high priority. I will add a switch for this. DockCtrl framework is (you may have noticed already) very flexible and it's easy to implement this

I know why! I know why I thought qt has dragable tabs That's because half a day I spend with eclipse.. And eclipse works the way I described as a solution to "satisfy us all". And I personally think this is the best option. It's very intuitive and you don't have to press any key to drag the tab. if you move a mouse outside tabs area you start to drag the whole window. Check this please (netbeans should work in the same way).
Quote:
What about implementing Ctrl OR Middle Mouse Button Drag ?

Yes it can be an option. I think we shoud make docking system as configurable as possible at least in this case.

PS: I also noticed a cool thing about qt. If dragged window is to be tabbed the destination place is not only light blue (like in our docking engine) but this blue area is also transparent so you can see previous tab content. I like it.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Thu, 21 Feb 2008 20:04:43 GMT
View Forum Message <> Reply to Message

mrjt wrote on Thu, 21 February 2008 13:27If you want to have a play with how I think the tabs should work I've attached my latest stable build. I've just finished implementing theming (Thanks Oblivion!), and nested tabs. Hold down CTRL while dragging a tab collection to add it as nested (there is a switch to change the default behaviour but I haven't added to the example since I'll have to make a better on soon anyway).

Ctrl is not bad. Although in QuickTabs I used Ctrl to scroll tabs. I still think it's better (as option or as default) to follow eclipse solution. If you start draging the tab you drag the tab until mouse left tabs area.
Quote:
I've also enabled tab grouping, with grayed-out tabs showing inactive (I find quicktabs just hiding them to be a bit confusing).

It looks quite nice, but the main idea behind grouping was to reduce the scrollbar.. and make switching between tabs more comfortable.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Thu, 21 Feb 2008 20:31:23 GMT
View Forum Message <> Reply to Message

Quote:
I know why! I know why I thought qt has dragable tabs Smile That's because half a day I spend with eclipse.. And eclipse works the way I described as a solution to "satisfy us all". And I personally think this is the best option. It's very intuitive and you don't have to press any key to drag the tab. if you move a mouse outside tabs area you start to drag the whole window. Check this please (netbeans should work in the same way).

Ok.

Quote:
Yes it can be an option. I think we shoud make docking system as configurable as possible at least in this case.

Well, I have good news for "us all" then . I've implemented the both types of dragging (Ctrl + left drag OR solely middle drag tab repositioning) for both Tabwindows and Autohide (As I said before, DockCtrl is highly and easily configurable ) and updated the Example exe and SVN Just check out. I want to know your opinion. I will make it configurable through a variable too.

Quote:
PS: I also noticed a cool thing about qt. If dragged window is to be tabbed the destination place is not only light blue (like in our docking engine) but this blue area is also transparent so you can see previous tab content. I like it.

Yes, that's cool but how are we going to paint transparent? I'm not very good in using Draw functions of U++. Do you have any idea? I'm not sure -- I didn't try yet -- but, could we use Ctrl::DrawCtrl(Draw& w, int x, int y) method to draw the previous tab content on the animation Ctrl which provides that effect (since animation ctrl is only a ctrl) ? I will give it a try ( maybe I could take a snapshot manually and process it with dithering or stg. else). It sounds reasonable to me.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Fri, 22 Feb 2008 13:33:52 GMT
View Forum Message <> Reply to Message

Quote:PS: I also noticed a cool thing about qt. If dragged window is to be tabbed the destination place is not only light blue (like in our docking engine) but this blue area is also transparent so you can see previous tab content. I like it.
I had also been thinking about this, but I've only just worked out how to do it (my internal structure has now got to a point where I modify stuff quite easily. Thank FSM). See attached for an example. I'm going to improve the tab behaviour I think, but that requires more changes that I'm not quite ready to make. Looks nice though

Oblivion: The trick is to overload Ctrl::PostPaint in you container class (TabWindow?) and then Draw something transparent over the whole View area. I'm using the following code to generate my highlight images:

```
Image StandardHighlight(Color inside, Color border)
{
 Size sz(5, 5);
 ImageBuffer ib(sz);
 RGBA *q = ~ib;
 for (int i = 0; i < 5; i++)
  for (int j = 0; j < 5; j++)
   *(q++) = (i == 4 || j == 4 || !i || !j) ? border : inside;
 ib.SetHotSpot(Point(1, 1));
 ib.Set2ndSpot(Point(3, 3));
 return ib;
}

Image AlphaHighlight(const Image &img, int alpha)
{
 ImageDraw draw(img.GetSize());
 draw.Alpha().DrawRect(img.GetSize(), Color(alpha, alpha, alpha));
 draw.DrawImage(0, 0, img);
 // Is there a better way to set hotspots than this?
 ImageBuffer ib((Image)draw);
 ib.SetHotSpot(Point(1, 1));
 ib.Set2ndSpot(Point(3, 3));
 return ib;
```

```
}

CH_STYLE(DockWindow, Style, StyleDefault)
{
 Image img = StandardHighlight(Blend(SColorHighlight, SColorPaper, 90), SColorHighlight);
 stdhighlight = img;
 tabhighlight = AlphaHighlight(img, 160);
}
```

unodgs wrote on Thu, 21 February 2008 20:04
Ctrl is not bad. Although in QuickTabs I used Ctrl to scroll tabs. I still think it's better (as option or as default) to follow eclipse solution. If you start draging the tab you drag the tab until mouse left tabs area.

I've just tried this and I don't think it possible with the current Upp Drag/Drop implementation. To make it work you need to call PasteClip::Reject from DragLeave. You have to do a nasty bodge just to get the PasteClip object to begin with, and calling Reject doesn't seem to release the mouse, so it's not possible to start dragging the window. The alternative is to not use DnD for tab dragging, but I'd much rather have a 'correct' solution.

Having tried it I agree that this is how it should work though, so I'll keep trying.

unodgs wrote on Thu, 21 February 2008 20:04
It looks quite nice, but the main idea behind grouping was to reduce the scrollbar.. and make switching between tabs more comfortable.
The grayed-out tabs are not included in the scroll limit. It works exactly the same as before except that you can see that there are non-active tabs.

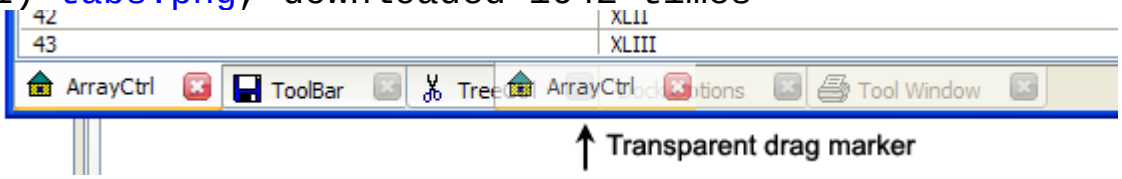
I'm also trying out a new drag highlight:

I think it's an improvement, what do you think?

Incidentally, all these features (grouping, scrollbar/tabbar autohide, etc.) are in my general TabBar class, based around standard Upp Value/Display architecture. QuickTabs is just a specialized bas class of this TabBar. It seems a shame to waste it on one project, should I release it to the SVN Bazaar?

## File Attachments

1) tabs.png, downloaded 1642 times




2) DockTest.zip, downloaded 431 times

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 22 Feb 2008 13:57:02 GMT
View Forum Message <> Reply to Message

Quote:
Oblivion: The trick is to overload Ctrl::PostPaint in you container class (TabWindow?) and then Draw something transparent over the whole View area. I'm using the following code to generate my highlight images:

Thank you
I'll implement this asap.

Quote:
I'm also trying out a new drag highlight:
index.php?t=getfile&id=1019&private=0
I think it's an improvement, what do you think?

It sure is an improvement.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Fri, 22 Feb 2008 14:20:10 GMT
View Forum Message <> Reply to Message

mrjt wrote on Fri, 22 February 2008 08:33
I had also been thinking about this, but I've only just worked out how to do it (my internal structure has now got to a point where I modify stuff quite easily. Thank FSM). See attached for an example. I'm going to improve the tab behaviour I think, but that requires more changes that I'm not quite ready to make. Looks nice though

Oblivion: The trick is to overload Ctrl::PostPaint in you container class (TabWindow?) and then Draw something transparent over the whole View area. I'm using the following code to generate my highlight images:

Well, I'm impressed. Transparent looks great! And I must say window animation is working very well too.
The only thing you should do is to update quicktabs paint routines. The one you have do not work well with all themes and in right and bottom case eats last line of tab. But wait until tomorrow. I will update quicktabs one more time.
Quote:
Having tried it I agree that this is how it should work though, so I'll keep trying.

Great to hear that
Quote:
I'm also trying out a new drag highlight:

I think it's an improvement, what do you think?

Yes, but in this case I think that moved tab should not only be painted in drop places but in every mouse position.
Quote:
Incidentally, all these features (grouping, scrollbar/tabbar autohide, etc.) are in my general TabBar class, based around standard Upp Value/Display architecture. QuickTabs is just a specialized bas class of this TabBar. It seems a shame to waste it on one project, should I release it to the SVN Bazaar?

Ok, no problem. As I said many times before we should have one tab system that could be easily plugged into your, Oblivion or anyone else docikng code or used as a standalone widget.
So please make it separate directory and put there your tabs code. I'll try to integerate it with my recent changes. And then we should somehow force oblivions code to use it. I think it's possible.

BTW: I think we should focus on one docikng system. As the Oblivion was first he's now officaly resonsible for docking system. But I wouldn't like to waste your achievements. Maybe you could somehow cooperate with Oblivion. I know that everyone of us prefers own solutions the most  but maybe after some changes/adding some fatures you could use oblivion system in your apps and then just start working at oblivion's (upp) code.
What you and Onlivion think about it. Oblivion, could be the work split somehow so you could work together?
Of course I'm not forcing to making one docking system but that would be ideal solution IMO.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Fri, 22 Feb 2008 14:26:40 GMT
View Forum Message <> Reply to Message

Oblivion, I think you should abandom DockCtrl DEV802b.2 and just move out DockCtrl and DoctCtrlExample to the root (like before).
For example I used svn root directory in upp assembly definition and I'm not able to run DockCtrlExample app because it has #include <DockCtrl/DockCtrl.h>. It's very ok, but in current directory layout I'm forced to chage it to <DockCtrl DEV802b.2/DockCtrl/DockCtrl.h>.
And it makes preparing instlation of upp releases much harder. Can we get back to previous layout, please?

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 22 Feb 2008 14:31:46 GMT
View Forum Message <> Reply to Message

Quote:
Oblivion, I think you should abandom DockCtrl DEV802b.2 and just move out DockCtrl and DoctCtrlExample to the root (like before).
For example I used svn root directory in upp assembly definition and I'm not able to run

DockCtrlExample app because it has #include <DockCtrl/DockCtrl.h>. It's very ok, but in current directory layout I'm forced to chage it to <DockCtrl DEV802b.2/DockCtrl/DockCtrl.h>.
And it makes preparing instlation of upp releases much harder. Can we get back to previous layout, please?

I guess you are right. My intention was to keep the SVN/Bazaar clean.  Ok, I'm moving it both to the main tree.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Fri, 22 Feb 2008 14:48:08 GMT
View Forum Message <> Reply to Message

unodgs wrote on Fri, 22 February 2008 14:20
Well, I'm impressed. Transparent looks great! And I must say window animation is working very well too.
The only thing you should do is to update quicktabs paint routines. The one you have do not work well with all themes and in right and bottom case eats last line of tab. But wait until tomorrow. I will update quicktabs one more time.

My TabBar Paint routine not based on the current QuickTabs Paint code, as I wrote it before you added support for different alignments. Mine works by pre-caching all the rotated/mirrored style elements so that you don't have to do image copying and rotation in the Paint routine. On the other-hand the code is more complex.

I also orient left-hand tabs in an (IMO) more natural way (Autohide something to the left in my example to see the difference).
Quote:Of course I'm not forcing to making one docking system but that would be ideal solution IMO.
I need to decide whether I really need the more complex form of Docking (not just single row, but nested tree of splitters). This was the original reason for needing my version, and it's not something that can be added at a later date.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 22 Feb 2008 15:14:04 GMT
View Forum Message <> Reply to Message

Quote:
So please make it separate directory and put there your tabs code. I'll try to integerate it with my recent changes. And then we should somehow force oblivions code to use it. I think it's possible.

No problem, besides I'll give it a try to implement that transparent tab dragging too.

Quote:
BTW: I think we should focus on one docikng system. As the Oblivion was first he's now officaly resonsible for docking system. But I wouldn't like to waste your achievements. Maybe you could somehow cooperate with Oblivion. I know that everyone of us prefers own solutions the most Smile but maybe after some changes/adding some fatures you could use oblivion system in your apps and then just start working at oblivion's (upp) code.
What you and Onlivion think about it. Oblivion, could be the work split somehow so you could work together?
Of course I'm not forcing to making one docking system but that would be ideal solution IMO.

Well, we are "unofficially" cooperating already (at least, I'm using some part of Mrjt's code. E.g. I took the Autohidebar animation control, X11 support (by the way, I've implemented mrjt's X11 support but have no time to test it due to some technical reasons. I Will upload it tonight to the SVN) alpha blending from him. (I've implemented this too, I took severeal lines of code ) And I'm very thankful to him.

But there is one big problem with our docking frameworks: they are, totally (or, let's say mostly) different. We use different techniques to handle the docks. They look similar, but not act so. So, IMHO, it would be painful to "merge" them. Yet we still can and already cooperate (As I said before, he helps me most of the time).

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 22 Feb 2008 15:23:27 GMT
View Forum Message <> Reply to Message

Quote:
Oblivion: The trick is to overload Ctrl::PostPaint in you container class (TabWindow?) and then Draw something transparent over the whole View area. I'm using the following code to generate my highlight images:

By the way, what is Ctrl::PostPaint()?
Is there such a method in Ctrl class ?

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Fri, 22 Feb 2008 15:27:04 GMT
View Forum Message <> Reply to Message

Oops!

That's an addition I made to my version ages ago. I rediscovered it this morning and forgot that I'd added it myself

And now I remember that it breaks things. Bugger.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 23 Feb 2008 02:14:37 GMT
View Forum Message <> Reply to Message

Ok, DND transparency effect and X11 DND support added to the source code. I've also reverted the DockCtrl SVN folder back to the bazaar/DockCtrl and Bazaar/DockCtrlExample.

And as usual, I aupdated the example exe. You can see the tranparency effect. (Thanks to Mrjt for the transparent image painting code)

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Sat, 23 Feb 2008 12:38:25 GMT
View Forum Message <> Reply to Message

I tried to compile SVN rev157 under FreeBSD and fixed some minor errors, but I got stuck with a missing call to MoveEnd():
/home/masu/upp_uvs/bazaar/DockCtrl/DockWindow.cpp: In member function 'virtual void DockWindow::EventPr
oc(Upp::Ctrl::XWindow&, XEvent*)':
/home/masu/upp_uvs/bazaar/DockCtrl/DockWindow.cpp:184: error: 'MoveEnd' was not declared in this scope
DockableCtrl.cpp
/home/masu/upp_uvs/bazaar/DockCtrl/DockableCtrl.cpp: In member function 'void DockableCtrl::StartWindow
Drag()':
/home/masu/upp_uvs/bazaar/DockCtrl/DockableCtrl.cpp:263: error: 'p' was not declared in this scope

I could not find this function inside project sources, but there is one in James' Docking package. Did you forget this one?

Matthias

## File Attachments

1) Dock.zip, downloaded 490 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 23 Feb 2008 13:22:05 GMT
View Forum Message <> Reply to Message

Quote:
I tried to compile SVN rev157 under FreeBSD and fixed some minor errors, but I got stuck with a missing call to MoveEnd():

/home/masu/upp_uvs/bazaar/DockCtrl/DockWindow.cpp: In member function 'virtual void DockWindow::EventPr
oc(Upp::Ctrl::XWindow&, XEvent*)':
/home/masu/upp_uvs/bazaar/DockCtrl/DockWindow.cpp:184: error: 'MoveEnd' was not declared in this scope
DockableCtrl.cpp
/home/masu/upp_uvs/bazaar/DockCtrl/DockableCtrl.cpp: In member function 'void DockableCtrl::StartWindow
Drag()':
/home/masu/upp_uvs/bazaar/DockCtrl/DockableCtrl.cpp:263: error: 'p' was not declared in this scope

I could not find this function inside project sources, but there is one in James' Docking package. Did you forget this one?

Ooops
I forgot to adjust the code(MoveEnd() should be removed).
Ok, I've updated the SVN. But there are two issues you need to know about the current SVN version.

1. The code is not tested. For, the computer I have installed the Ubuntu distro was broken. I'll get my hands on it this week. So I need some feedback about the DnD behaviour. I would be grateful if you could give me some feedbacks about the DockCtrl X11 behaviour.

2. Tab transparency effect has a strange bug which causes a tab window in a PaneFrame with more than one childs to draw improperly(I'm trying to find the reason of this strange behaviour).

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Sat, 23 Feb 2008 21:35:51 GMT
View Forum Message <> Reply to Message

Unfortunately, the application does not start up.
I attached an archive with a patch (has to be applied in DockCtrl directory) that contains once

more the changes I had to do to get the package compiled.
The archive also contains the buglog and log files, an error occurred during Value conversion (at least that's what an error dialog told me).

Matthias

## File Attachments

1) DockCtrl.zip, downloaded 483 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sat, 23 Feb 2008 23:22:41 GMT
View Forum Message <> Reply to Message

Well, that's a bad news. Whatever, I'm growing impatient. Hopefully I'll have the chance to examine it this week (I'll get my mobo back from the computer service on monday. After then, my primary target will be X11 support...

Thank you for your feedback. I'll apply any changes necessary.

Also, the tab transparency effect bug is fixed. It works fine on (2000/XP Pro and Home/Vista - Basic).

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 24 Feb 2008 00:18:22 GMT
View Forum Message <> Reply to Message

Ok, Masu I've patched the files. Also, I've re-installed that ever-complaining cr... compiler (MingW 5.1.3) again. It seems that the crash was due to a typecasting bug in AlphaHighlight() function.

```
static Image AlphaHighlight(const Image &img, int alpha)
{
 ImageDraw draw(img.GetSize());
 draw.Alpha().DrawRect(img.GetSize(), Color(alpha, alpha, alpha));
 draw.DrawImage(0, 0, img);
 // MingW (5.1.3) fix - Damn, I hate that ever-complaining compiler.
 Image dimg = draw;
 // Is there a better way to set hotspots than this?
 ImageBuffer ib(dimg);
 ib.SetHotSpot(Point(1, 1));
 ib.Set2ndSpot(Point(3, 3));
 return ib;
}
```

Mrjt, you should check this out too.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Sun, 24 Feb 2008 14:34:15 GMT
View Forum Message <> Reply to Message

Unfortunately, the error persist.
I began to debug application and found that the error somehow occurs when trying to cast from 'const Value::Void*' to 'const RichValueRep' that appears in Value.h line 438.

Matthias

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Sun, 24 Feb 2008 15:11:21 GMT
View Forum Message <> Reply to Message

Ok, I have found the problem.
In DockCtrl.h change line 634 from
DragBar() : _parent(0), _titlesize(0) { SetFrame(FieldFrame()); }

to
DragBar() : _parent(0), _titlesize(0,0) { SetFrame(FieldFrame()); }

Size constructor parameter initialization was wrong and therefore RichValue conversion to Size_ failed.

So, now I can start application, but when trying to drag first frame, it is only detached and not movable until I release the mouse button. After that action the GUI is not refreshed anymore.
Also the auto hide feature does not work, i.e. when I move the mouse cursor over Dock 6 it is highlighted, but the window does not appear.
What however works is Tab position dragging.

Matthias

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Sun, 24 Feb 2008 16:44:05 GMT
View Forum Message <> Reply to Message

Quote:
Size constructor parameter initialization was wrong and therefore RichValue conversion to Size_ failed.

So, now I can start application, but when trying to drag first frame, it is only detached and not movable until I release the mouse button. After that action the GUI is not refreshed anymore. Also the auto hide feature does not work, i.e. when I move the mouse cursor over Dock 6 it is highlighted, but the window does not appear.
What however works is Tab position dragging.

Matthias

Thank you very much. I'll change that. Please be patient (so should I ) Tomorrow when I get my mobo back I'll focus exclusively on these issues.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Mon, 25 Feb 2008 09:35:30 GMT
View Forum Message <> Reply to Message

The problem with the AutoHide is that X11 doesn't open the popup properly if any of the dimensions are 0. I fixed this after Oblivion copied my code.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Mon, 25 Feb 2008 18:32:12 GMT
View Forum Message <> Reply to Message

Quote:
The problem with the AutoHide is that X11 doesn't open the popup properly if any of the dimensions are 0. I fixed this after Oblivion copied my code.

Yes, I've noticed that later, Thank you mrjt.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Sc0rch on Tue, 26 Feb 2008 00:05:33 GMT
View Forum Message <> Reply to Message

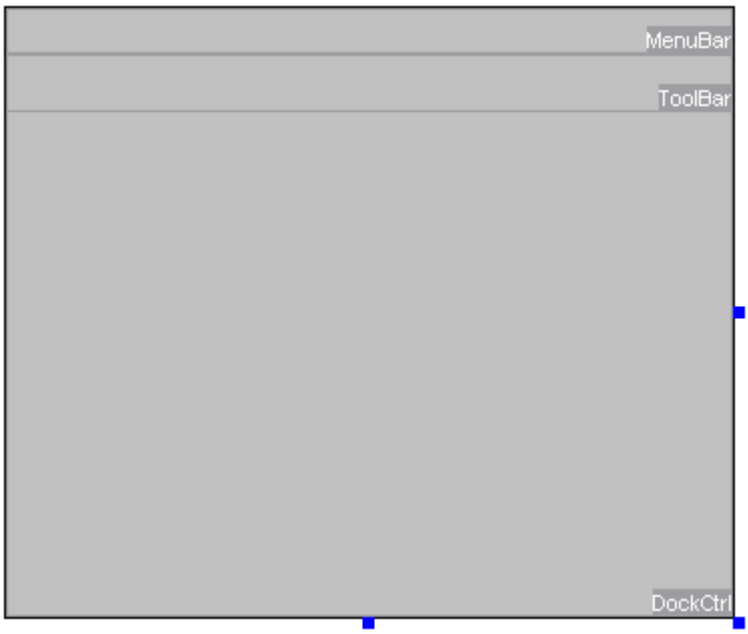
Please, sorry for my English (just learning). Screenshot below illustrates how I want to setup widget areas. But whatever I do, DockCtrl overrides content or/and sets it into the center of the window.

Please, help with a little code-snippet or a little bit modified DockCtrlExample or tell me in what direction I have to dig.

Thank you,
Anton.

## File Attachments

1) image1.png, downloaded 1006 times




---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 26 Feb 2008 01:14:38 GMT
View Forum Message <> Reply to Message

Quote:
Please, sorry for my English (just learning). Screenshot below illustrates how I want to setup widget areas. But whatever I do, DockCtrl overrides content or/and sets it into the center of the window.

Please, help with a little code-snippet or a little bit modified DockCtrlExample or tell me in what direction I have to dig.

Thank you,
Anton.

Yes, that's because I didn't add a barctrl check mechanism to the DockCtrl (it's on my todo list). Nevertheless, the solution to your problem is very easy, just add any barctrl (menu, toolbar) to the main window you use BEFORE Add()ing DockCtrl.

I've added the DockCtrlExample below, take a look at the lines 26-29 in main.cpp. I hope it'll help

you.

And please keep in mind that DockCtrl in not in production state. It still has a long way to go (But Hopefully, it'll be mature enough at the end of March).

## File Attachments

1) DockCtrlExample.rar, downloaded 420 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by ScOrch on Tue, 26 Feb 2008 05:58:12 GMT
View Forum Message <> Reply to Message

Oblivion
Yes, that's because I didn't add a barctrl check mechanism to the DockCtrl (it's on my todo list).

Yes, of course, I understand, the screenshot was just an illustration of idea.

Oblivion
Nevertheless, the solution to your problem is very easy... I hope it'll help you.

I'm using Upp just a month and I've completely forgot about AddFrame-methods. Menubar's behaviour (as popup-menu) looks confusing me, but I'll read a appropriate threads. Thank you.

Oblivion
And please keep in mind that DockCtrl in not in production state. It still has a long way to go (But Hopefully, it'll be mature enough at the end of March).

Of course, it still has a way to go. But current state of DockCtrl development also very usefull. I'm looking for use it in my projects.

Here the latest t-file from SVN-repository, completely translated to Russian, and more correct layout-file. I hope that it can be usefull for you.

Anton

## File Attachments

1) DockCtrl.t, downloaded 516 times
2) DockCtrl.lay, downloaded 507 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 26 Feb 2008 09:41:53 GMT
View Forum Message <> Reply to Message

Quote:
Here the latest t-file from SVN-repository, completely translated to Russian, and more correct layout-file. I hope that it can be usefull for you.

Anton

Thank you very much. I'll add these to the current SVN today.

Regards

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 26 Feb 2008 23:41:56 GMT
View Forum Message <> Reply to Message

Finally, I've managed to fix all the MAJOR issues about X11. DockCtrl now has full docking support on X11 platform. Tested on Kubuntu 7.10 (Ubuntu with KDE) and works perfectly.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by unodgs on Wed, 27 Feb 2008 09:13:25 GMT
View Forum Message <> Reply to Message

Quote:Well, we are "unofficially" cooperating already (at least, I'm using some part of Mrjt's code. E.g. I took the Autohidebar animation control, X11 support (by the way, I've implemented mrjt's X11 support but have no time to test it due to some technical reasons. I Will upload it tonight to the SVN) alpha blending from him. (I've implemented this too, I took severeal lines of code ) And I'm very thankful to him.
I can see that That's really great!
Quote:
But there is one big problem with our docking frameworks: they are, totally (or, let's say mostly) different. We use different techniques to handle the docks. They look similar, but not act so. So, IMHO, it would be painful to "merge" them. Yet we still can and already cooperate (As I said before, he helps me most of the time).

That's a pity but I must check what the differences really are. The point is API can be different but as long as everything mrjt want can be done in our API there is no reason to develop his own docking system. It's better to extened the existing one.
On the second hand if that's ok for mrjt that we use his code to improve our system let's stay with current develop model.
Quote:
My TabBar Paint routine not based on the current QuickTabs Paint code, as I wrote it before you added support for different alignments. Mine works by pre-caching all the rotated/mirrored style

elements so that you don't have to do image copying and rotation in the Paint routine. On the other-hand the code is more complex.

Yes, I'm gonna do the same, but later. For now I just want tabs to paint and behave corretly.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Wed, 27 Feb 2008 09:37:37 GMT
View Forum Message <> Reply to Message

Quote:On the second hand if that's ok for mrjt that we use his code to improve our system let's stay with current develop model.
That's fine. I've decided to drop the insanely complicated docking, but since I'm almost 'finished' I'm going to develop mine a bit further as it's much easier for me to experiment with features there, and it's already in use in an App.

Quote:Quote:My TabBar Paint routine not based on the current QuickTabs Paint code, as I wrote it before you added support for different alignments. Mine works by pre-caching all the rotated/mirrored style elements so that you don't have to do image copying and rotation in the Paint routine. On the other-hand the code is more complex.
Yes, I'm gonna do the same, but later. For now I just want tabs to paint and behave corretly.
Cool, it's certainly easier that way. If you can perfect the standard version I can then incorporate the changes, and at some point today I'll upload my TabBar to the SVN and you can have a look.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 27 Feb 2008 11:57:33 GMT
View Forum Message <> Reply to Message

Quote:
That's a pity but I must check what the differences really are. The point is API can be different but as long as everything mrjt want can be done in our API there is no reason to develop his own docking system. It's better to extened the existing one.
On the second hand if that's ok for mrjt that we use his code to improve our system let's stay with current develop model.

Well, there is no problem, except for the "complex docking". Of course I can add that feature to the DockCtrl too, but it will 1) bloat the code, 2) make my life harder , and frankly I'm not sure that it is a must, for we already have some different features like nested tabbing, nested autodocking and even a nested dockable framework. As to the other features, we already have most of them. If he permits, we sure can use James' tab code (that highligted tabs are really cool) and other new ideas he could give us. He could also make changes on the current DockCtrl SVN (James, if you have time, you are permitted "officially" to make any changes you see

necessary.)

I can make "complex docking" an option, but probably in the next major version. Because, I need to change the internals without effecting the user API (and I don't wan't to make the current code instable).

By the way, does anyone have any idea about this issue:

http://www.ultimatepp.org/forum/index.php?t=msg&th=3220& amp;start=0&

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Sc0rch on Wed, 27 Feb 2008 21:37:12 GMT
View Forum Message <> Reply to Message

Is it possible to call the context-menu with panel-settings  by right-clicking on the panel's title (not by button)?

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Wed, 27 Feb 2008 22:20:43 GMT
View Forum Message <> Reply to Message

Quote:
Is it possible to call the context-menu with panel-settings by right-clicking on the panel's title (not by button)?

Firstly, which "panel"? There is only one panel and it is the control panel which you can control the behaviour and change settings of the framework. In control panel, if you want to invoke context menu of a specific dock, just left click on the name of the dock in the item list. But if you mean "docks" in the main work area, I can say that it is very easy to add such behaviour. I can add that if it is really needed but what will be the use (no rhetoric, this is a question I'm asking you) ?

In normal dockwindows you always see the "title bar" so the context menu button will be always there for you to push. In tabwindows, you can right click on the tab to get the context menu of specific tabbed window. AND in floating mode, the behaviour you've just described is already implemented. Just right click the window NC area (bar) to get context menu (if in "floating" mode).

If you (or anyone) think it will be really useful, I will add it to the current code.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Sc0rch on Thu, 28 Feb 2008 13:48:20 GMT
View Forum Message <> Reply to Message

OblivionBut if you mean "docks" in the main work area, I can say that it is very easy to add such behaviour.
Sorry, please, of course I mean "docks".

OblivionI can add that if it is really needed but what will be the use (no rhetoric, this is a question I'm asking you) ?
For example, I want to hide menu and close buttons, but have an opportunity to access advanced options and commands by using the context-menu of the dockwindow's title. And also it will be great if you'll add something like a WhenMenuBar-callback in upp-widgets for user's context-menu.

OblivionIf you (or anyone) think it will be really useful, I will add it to the current code.
It will be good for me, but it was just a wish/question and if you think that such little things can take a time to wait, I'll understand you and, of course, I can wait for such a little opportunities to change something for a long time.

I hope that you'll can understand me, =). Sometimes it's hard to explain all, but I'm trying.

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Thu, 28 Feb 2008 14:00:08 GMT
View Forum Message <> Reply to Message

Quote:

For example, I want to hide menu and close buttons, but have an opportunity to access advanced options and commands by using the context-menu of the dockwindow's title.

Well, that sounds nice. Since we want the DockCtrl to be as configurable as possible while keeping it as simple as it can be for the user, ok, I've added it to my todo list. Let's make it an option both for the user (via settings/control panel) and for the developer (via API).

Quote:
And also it will be great if you'll add something like a WhenMenuBar-callback in upp-widgets for user's context-menu.

I'm not sure if I understand this. Correct me if I'm wrong. You're asking me to add a callback to the docks which gets called (a callback with or without any variable to pass?) when context menu is invoked. Right? If so, ok I'll add it. If not, could you explain the behaviour of the callback you requested a bit?

And please don't get me wrong, I'm not a member of Spanish Inquisition, I'm just asking these questions because I need rationale to add these extensions/changes.

Also, thank you for your feedback. As always is, any suggestion, feedback, bug report and constructive criticism is always welcome.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Sc0rch on Fri, 29 Feb 2008 01:22:58 GMT
View Forum Message <> Reply to Message

Oblivion
Well, that sounds nice. Since we want the DockCtrl to be as configurable as possible while keeping it as simple as it can be for the user, ok, I've added it to my todo list.
Great to hear, thank you.

OblivionYou're asking me to add a callback to the docks which gets called (a callback with or without any variable to pass?)  when context menu is invoked. Right?
Yes, like in a GridCtrl, for example.

Oblivion
And please don't get me wrong, I'm not a member of Spanish Inquisition, I'm just asking these questions because I need rationale to add these extensions/changes.
Do not worry, all is good =). And other questions:

1) Do you can add an opportunity to switch-off the automatic-docking of the dock-window while dragging? Can you add such switch to each dockwindow? That will be a great feature for us and our projects, I think. And floating windows will be more easy to use in my case.

2) What about a special dockwindow for toolbar? There are images below illustrating the idea.

OblivionAlso, thank you for your feedback. As always is, any suggestion, feedback, bug report and constructive criticism is always welcome.
I'm just a newbie for now, but I keep an eye on your project, and can translate it if needed.

P.S. I noticed that "Autohide" command is enabled in floating dock-window's menu, but in floating mode it is not useful. Probably, you can disable this command in floating mode.

Thank you,
Anton

## File Attachments

1) float1.png, downloaded 1008 times




2) float2.png, downloaded 1231 times




3) docked1.png, downloaded 1230 times



4) docked2.png, downloaded 1228 times



---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 29 Feb 2008 06:21:12 GMT
View Forum Message <> Reply to Message

Quote:
1) Do you can add an opportunity to switch-off the automatic-docking of the dock-window while dragging? Can you add such switch to each dockwindow? That will be a great feature for us and our projects, I think. And floating windows will be more easy to use in my case.

Yes this could be an option (but let's add it only to the developer API, the control panel is mainly

for the users, so it should stay simple)

Quote:
2) What about a special dockwindow for toolbar? There are images below illustrating the idea.

Well, in the DockCtrl.h (DockableCtrl class), there is:

enum { TYPE_WINDOCK, TYPE_TABDOCK, TYPE_BARDOCK };

So, a dockable bar is already planned and in my todo list. That's why the base class of all DockCtrl widgets is DockableCtrl. I'll implement a DockBar class (since, it's relatively easy to add it) after I fix all the major issues of the main framework.

Quote:
P.S. I noticed that "Autohide" command is enabled in floating dock-window's menu, but in floating mode it is not useful. Probably, you can disable this command in floating mode.

Thanks, I'll fix it asap.

Quote:
I'm just a newbie for now, but I keep an eye on your project, and can translate it if needed.

Yes, please. I need translators and feedbacks all the time.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Fri, 29 Feb 2008 16:18:45 GMT
View Forum Message <> Reply to Message

I've just updated my Docking package. I've changed my Dock to a normal Splitter, so the animation code might be useful.

The TabBar is now uploaded too.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 21 Mar 2008 22:24:33 GMT

View Forum Message <> Reply to Message

Hi, I was away for a while. But now I'm back with the new version of DockCtrl.

This version has several improvements (mostly, stability issues fixed). Full serialization and custom/default layout support is added -- See ChangeLog.txt for details. Some API changes have been made. AS always, you can find the example exe in the first post of this topic. And from now on, I also upload the Source code of DockCtrl (You don't have to go the U++ SVN anymore) here too. However, you can find the nighly builds only on SVN.

You should use NOGTK keyword to compile DockCtrl correctly on KDE . Since, there was (or is?) a heap leak problem with the U++ core).

for more details about the problem, see topic:

http://www.ultimatepp.org/forum/index.php?t=msg&th=3220& amp; amp; amp;start=0&

The problem seem to be fixed in the nightly builds of U++ but I Didn't have time to download it. So if you have the latest build, just remove the NOGTK keyword.

Adn one more thing: You should remove DockCtrl and DockCtrlExample folders from "DockCtrl Package" folder in order to compile them without errors.

Any bug reports, constructive criticism and feedbacks are welcome.

Regards.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Sc0rch on Sat, 22 Mar 2008 03:41:36 GMT
View Forum Message <> Reply to Message

Oblivion wrote on Fri, 21 March 2008 23:24Full serialization and custom/default layout support is added -- See ChangeLog.txt for details... And from now on, I also upload the Source code of DockCtrl (You don't have to go the U++ SVN anymore) here too.
Great to hear! Big steps to Perspectives, very usefull feature for a large project.

A little behaviour-idea: left click on tab's header (one or double) switchs the dockwindow from autohide-mode to normal-docked state.

Updated translation here:

File Attachments
1) DockCtrl.t, downloaded 459 times

---

Subject: Re: DockCtrl (A dockable window widget for U++)

Posted by Oblivion on Sat, 22 Mar 2008 11:59:21 GMT
View Forum Message <> Reply to Message

Sc0rch wrote on Sat, 22 March 2008 05:41

A little behaviour-idea: left click on tab's header (one or double) switchs the dockwindow from autohide-mode to normal-docked state.

Yes nice idea. Added.  Also, package + exe updated.

Thank you for the translation.

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by mrjt on Tue, 25 Mar 2008 13:07:26 GMT
View Forum Message <> Reply to Message

Very nice, your transparent highlights work well (I think I might use that code  ).

A few things:
- Besides the fact that panelicons would be better as an Array<ImageCtrl> (to avoid using new/delete), would it not be easier to just add the Image to the ArrayCtrl and set the display for that column to ImageDisplay()?
- It's a bit wierd that all the windows open as peer-windows, it would seem more natural to me for them to be child windows.
- Instead of newlayoutdialog, you could just use Upp::EditText.

James

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Tue, 25 Mar 2008 13:54:26 GMT
View Forum Message <> Reply to Message

Quote:
Very nice, your transparent highlights work well (I think I might use that code.

Be my guest

Quote:
Besides the fact that panelicons would be better as an Array<ImageCtrl> (to avoid using new/delete), would it not be easier to just add the Image to the ArrayCtrl and set the display for that column to ImageDisplay()?

Well, that was a remnant of an attempt to add some "eyecandy" which I found useless and abandoned later. But you are right, I removed all the heap allocation stuff, and in the new versions (wihch I'll upload tonight) only tab windows will use heap (maybe I can revert them too...).

Quote:
- It's a bit wierd that all the windows open as peer-windows, it would seem more natural to me for them to be child windows.

Ooops, shame on me!
I forgot to change that in the public version. Actually, I opened them that way beacuse in KDE it seems that tool windows cannot be resized (I'm not sure, I didn't have the time to examine the behaviour further). Then I found that opening them as child windows will solve the problems (at least no peer) but forgot to update source code from diff. file. It'll be changed tonight.

Quote:
instead of newlayoutdialog, you could just use Upp::EditText.

Thanks, changed. By the way, why don't write some documentation about U++'s undocumented features. From the forums, I can say that you really seem to know about them very well.

And belive me, if I had some comprehensive documents about U++ under my hand, this project would have been finished months ago. And as you may have noticed, I'm too used to MFC (that good old C++ API, which you have to find a "workaround" for ). So this project is also a "tutorial" for me. When I finish this, I'll write some documents about undocumented API functions of U++...

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Mar 2008 02:15:25 GMT
View Forum Message <> Reply to Message

Hi, DockCtrl is updated to DEV803b.2

Predefined/user widget grouping support and a group manager addded. (Though, I'm well aware that it needs fine tuning).

Adding a predefine group is just a piece of cake:

(the snippet below is taken from the DockCtrlExample source)

```
void DockCtrlExample::InitCustomLayouts()
{
 // This method is called by DockCtrl for predefined custom layouts/groups.
```

```
 // Predefined Group Example
 // ------------------------
 // You need to take two simple steps to add a predefined widget group to DockCtrl based application:
 // Step 1: Add the group using WidgetGroup() method.
 // Step 2: Use "%" operator to add ctrls to group.

 WidgetGroup("Predefined Group 1") % dock1 % dock2 % dock3 % dock4;
 WidgetGroup("Predefined Group 2") % dock5 % dock6 % dock7;

 // Predefined Layout Example
 // ------------------------
 // NOT IMPLEMENTED YET. TO BE WRITTEN (SOON)...
}
```

Some known minor bugs are fixed. DockCtrl is quite stable now (Or I hope so...)

Also I have added to the source package (which you can find in the first post of this topic) a UWord Example with DockCtrl support. Just a simple app. But it is more concrete than the DockCtrlExample. Please keep in mind that this app is for mainly for beta testing/bug reporting purpose... It does NOT show the full potential of DockCtrl.

DockCtrl Dev804x.x will be mostly a general code cleanup.

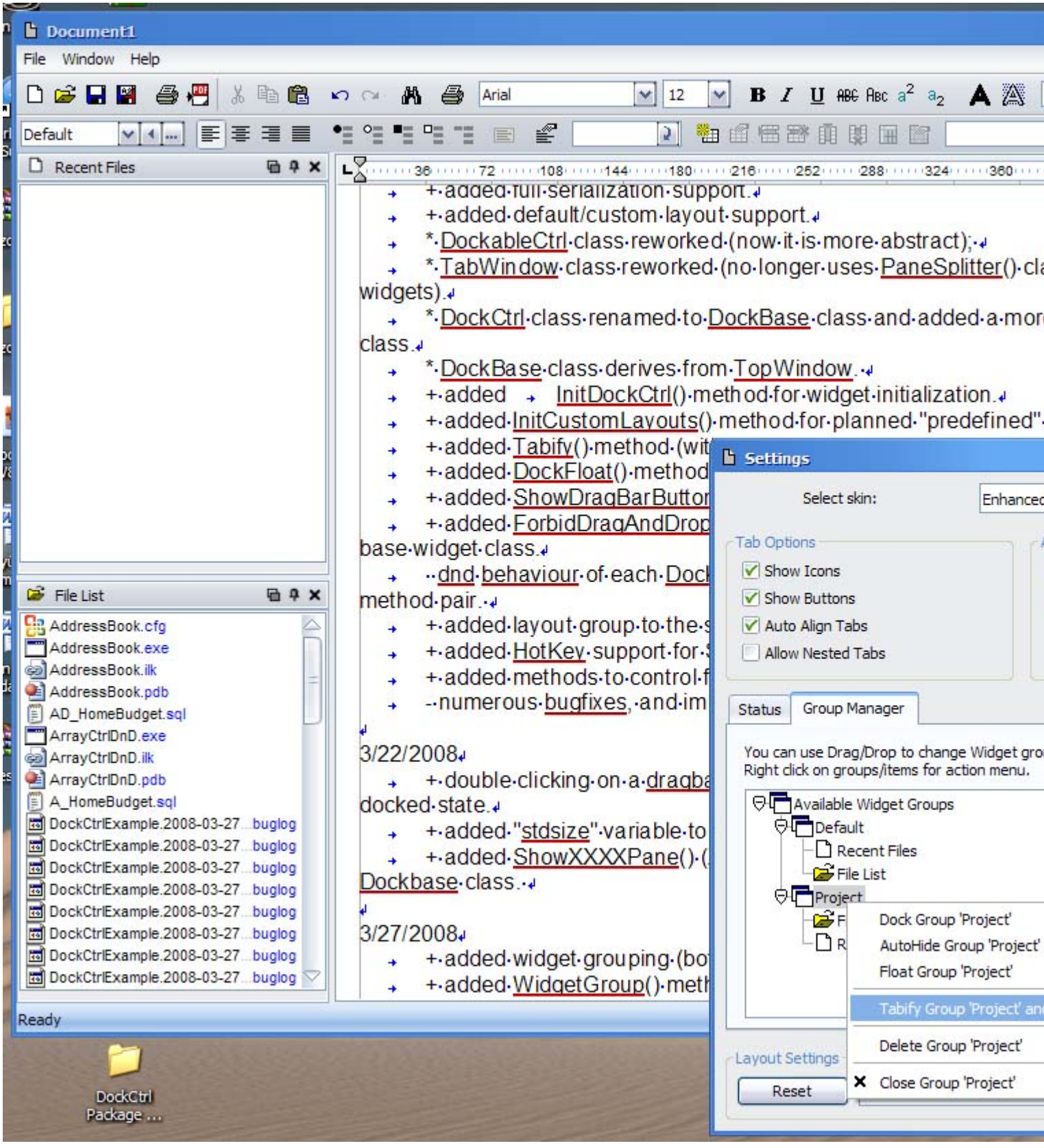
And, I have finally started to write a DockBar (dockable barctrl)  It will probably be released with (Dev805b.1)...

Any bug reports, constructive criticism and feedbacks are welcome.

Regards.

## File Attachments

1) UWord With DockCtrl.jpg, downloaded 1974 times


Document1
File Window Help
Arial
12
Default
Recent Files
File List
AddressBook.cfg
AddressBook.exe
AddressBook.ilk
AddressBook.pdb
AD_HomeBudget.sql
ArrayCtrlDnD.exe
ArrayCtrlDnD.ilk
ArrayCtrlDnD.pdb
A_HomeBudget.sql
DockCtrlExample.2008-03-27 buglog
Ready
DockCtrl Package ...
Settings
Select skin:
Enhanced
Tab Options
Show Icons
Show Buttons
Auto Align Tabs
Allow Nested Tabs
Status
Group Manager
You can use Drag/Drop to change Widget gro
Right click on groups/items for action menu.
Available Widget Groups
Default
Recent Files
File List
Project
Dock Group 'Project'
AutoHide Group 'Project'
Float Group 'Project'
Tabify Group 'Project' an
Delete Group 'Project'
Close Group 'Project'
Layout Settings
Reset

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Fri, 28 Mar 2008 13:52:51 GMT
View Forum Message <> Reply to Message

Good work!

However, what I have noticed is that undocked windows always stay in the foreground even when I activate another application's window. I would expect it to also be hidden.

Regards,
Matthias

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Mar 2008 14:39:39 GMT
View Forum Message <> Reply to Message

masu wrote on Fri, 28 March 2008 15:52Good work!

However, what I have noticed is that undocked windows always stay in the foreground even when I activate another application's window. I would expect it to also be hidden.

Regards,
Matthias

It is because, DockWindow derived classes are by default top most windows. Since a DockWindow is a TopWindow derived class, to change that behaviour all you have to do is to explicitly state that. e.g. TopWindow::TopMost(false).

For example, in DockCtrlExample constructor (or anywhere, at your will) you can;

dock1.TopMost(false);
dock2.TopMost(false);
etc...

However, You are right it is more traditional. So I'll change the default behaviour to TopMost(false).

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by masu on Fri, 28 Mar 2008 14:55:39 GMT
View Forum Message <> Reply to Message

I set all docked windows to TopMost(false) in main, but the tabbed undocked windows still stay in

the foreground.

Matthias

---

Subject: Re: DockCtrl (A dockable window widget for U++)
Posted by Oblivion on Fri, 28 Mar 2008 15:14:32 GMT
View Forum Message <> Reply to Message

masu wrote on Fri, 28 March 2008 16:55I set all docked windows to TopMost(false) in main, but the tabbed undocked windows still stay in the foreground.

Matthias

I'm sorry, I'm a little bit tired today...

Here is the quick fix:
In DockCtrl/DockWindow.cpp, line 35:

Sizeable().Zoomable().ToolWindow().TopMost().MinimizeBox().MaximizeBox().BackPaint();

Please change it to "TopMost(false)"

I've made the change and uploaded the modified ver. to the SVN.
Also, I will add some optional settings switchs for dock windows behaviour.

Regards.

---