

---

Subject: Docking host platform dependency refactored

Posted by [mirek](#) on Tue, 11 Jun 2013 06:17:37 GMT

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

---

Docking package had some platform dependent code (for Win32/X11) used to track moving of tool windows. This obviously created problems when porting to other platforms.

I have refactored this issue out by using frameless TopWindow and implementing moving behavior in our code. In the process I have also added frameless support for plain X11 and fixed some mouse event/capture related issues.

Mirek

---

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [Novo](#) on Tue, 25 Jun 2013 00:44:05 GMT

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

---

It looks like this refactoring changed the way docking panel can be resized. DockableCtrl can be resized independently from enclosing window. Example:

```
void wnb3::DockInit()
{
    DockableCtrl& target01 = Dockable(ont_tree, "Ontology Tree").SizeHint(Size(400, 300));
    DockRight(target01);

    Tabify(target01, Dockable(wtree, "Tree").SizeHint(Size(200, 300)));
    Tabify(target01, Dockable(wtext, "Text").SizeHint(Size(200, 300)));
}
```

I just dragged right and left sides of the DockableCtrl. Resizing of the main window restores correct layout of the DockableCtrl.

This happens in Windows and Linux.

---

#### File Attachments

1) [docking02.png](#), downloaded 1125 times

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [mirek](#) on Tue, 25 Jun 2013 06:49:43 GMT

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

---

I have to say I am not sure I understand the issue. Can you be more specific please? (eventually adding some test-case).

Mirek

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [Novo](#) on Tue, 25 Jun 2013 14:12:57 GMT

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

---

mirek wrote on Tue, 25 June 2013 02:49I have to say I am not sure I understand the issue. Can you be more specific please? (eventually adding some test-case).

Mirek

reference/DockingExample1

I can make this example look like below.

Changing size of the main window will fix the layout.

It is possible to resize individual DockingControls, which are supposed to be aligned to the main window. So, instead of changing size of the main window you end up changing size of a DockingControl. I do not know how else this can be explained. For example, ArrayCtrl 2 is docked to the top and it is supposed to be aligned to the left and right side of the main window, but I can resize it in any direction. Resizing the main window will relayout docking controls. This doesn't look like a consistent behavior.

#### File Attachments

1) [docking04.png](#), downloaded 948 times

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [mirek](#) on Wed, 26 Jun 2013 10:09:32 GMT

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

---

Ah, I see - the problem is that the "window" can be resized even if it is docked, right?

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [Novo](#) on Wed, 26 Jun 2013 13:22:07 GMT

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

---

mirek wrote on Wed, 26 June 2013 06:09Ah, I see - the problem is that the "window" can be resized even if it is docked, right?

I guess you are right.

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [mirek](#) on Wed, 26 Jun 2013 17:29:13 GMT

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

---

Novo wrote on Wed, 26 June 2013 09:22mirek wrote on Wed, 26 June 2013 06:09Ah, I see - the problem is that the "window" can be resized even if it is docked, right?

I guess you are right.

---

Hopefully fixed.

Mirek

---

---

**Subject: Re: Docking host platform dependency refactored**

Posted by [Novo](#) on Thu, 27 Jun 2013 16:52:35 GMT

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

---

mirek wrote on Wed, 26 June 2013 13:29Novo wrote on Wed, 26 June 2013 09:22mirek wrote on Wed, 26 June 2013 06:09Ah, I see - the problem is that the "window" can be resized even if it is docked, right?

I guess you are right.

Hopefully fixed.

Mirek

Thank you! It is fixed.

There is another issue, well, this is not a problem, that I can see in the new version of Docking: you cannot dock two floating windows into each other. You have to dock first floating window into the main window, and only after that you can dock another floating window into it. You can undock a bunch of tabified windows, after that you can undock individual tabified windows, but you cannot restore previous state by creating a floating windows with tabs. This is easily reproducible with DockingExample1.

Thanks.

---

---

**Subject: Re: Docking host platform dependency refactored**

Posted by [Novo](#) on Wed, 03 Jul 2013 04:16:46 GMT

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

---

Another observation: floating window has a weird dockbar. It has no menu and only one button with weird tooltip (Shift+65617246). Example: reference/DockingExmple1. Checked on Windows.

---

---

**Subject: Re: Docking host platform dependency refactored**

Posted by [mirek](#) on Wed, 03 Jul 2013 05:47:38 GMT

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

---

Thanks for reporting. I am trying to fix all of that...

Anyway, are you sure that "tabify against floating" really worked in previous versions? I have checked the code (albeit not thoroughly yet) and I see no support yet...

Mirek

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [281264](#) on Sat, 13 Jul 2013 10:04:23 GMT

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

---

When compiling DockingExample1 the compiler reports this error:

----- CtrlLib ( GUI MSC9 DEBUG WIN32 MSC ) (1 / 11)

----- Docking ( GUI MSC9 DEBUG WIN32 MSC ) (2 / 11)

DockPane.cpp

C:\Upp\Bazaar\Docking\DockPane.cpp(7) : error C2065: 'width' : undeclared identifier

C:\Upp\Bazaar\Docking\DockPane.cpp(400) : error C2065: 'width' : undeclared identifier

C:\Upp\Bazaar\Docking\DockPane.cpp(404) : error C2065: 'width' : undeclared identifier

Docking: 1 file(s) built in (0:01.25), 1250 msecs / file, duration = 1297 msecs, parallelization 0%

There were errors. (0:02.07)

Apparently, the variable width is not defined.

Javier

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [mirek](#) on Sat, 13 Jul 2013 19:17:06 GMT

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

---

This is really weird, are you sure that you have the latest version?

I have first tried my current local copy, then svn checkout, then downloaded sources from the nightly build and everything is OK.

Mirek

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [BioBytes](#) on Sat, 13 Jul 2013 19:25:19 GMT

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

---

Hi,

I got the same compiling issue with DockingExample1 (width variable is not defined!?):

----- CtrlLib ( GUI GCC DEBUG SHARED DEBUG\_FULL BLITZ LINUX POSIX ) (1 / 11)

----- Docking ( GUI GCC DEBUG SHARED DEBUG\_FULL BLITZ LINUX POSIX ) (2 / 11)  
BLITZ: DockWindow.cpp DockConfig.cpp DockMenu.cpp DockableCtrl.cpp DockCont.cpp  
DockPane.cpp DockTabBar.cpp  
----- CtrlCore ( GUI GCC DEBUG SHARED DEBUG\_FULL BLITZ LINUX POSIX ) (3 / 11)  
----- PdfDraw ( GUI GCC DEBUG SHARED DEBUG\_FULL BLITZ LINUX POSIX ) (4 / 11)  
BLITZ: TTFSStream.cpp TTFSStruct.cpp TTFRReader.cpp TTFSSubset.cpp  
PdfDraw.cpp  
PdfReport.icpp  
----- Draw ( GUI GCC DEBUG SHARED DEBUG\_FULL BLITZ LINUX POSIX ) (5 / 11)  
BLITZ: Font.cpp FontCR.cpp FontWin32.cpp FontFc.cpp Draw.cpp DrawText.cpp DrawData.cpp  
Drawing.cpp ImageAnyDraw.cpp DrawUtil.cpp DrawTextUtil.cpp Display.cpp Image.cpp  
ImageBlit.cpp RasterFormat.cpp RasterW  
rite.cpp Palette.cpp Raster.cpp RasterEncoder.cpp ImageOp.cpp ImageChOp.cpp  
ImageScale.cpp BiCubic.cpp RescaleFilter.cpp MakeCache.cpp DrawRasterData.cpp  
DDARasterizer.cpp SDrawClip.cpp SDrawPut.cpp SDrawTe  
xt.cpp SDraw.cpp SImageDraw.cpp Cham.cpp SSettings.cpp  
----- plugin/bmp ( GUI GCC DEBUG SHARED DEBUG\_FULL BLITZ LINUX POSIX ) (6 / 11)  
BLITZ: Bmp.cpp BmpWrite.cpp Icon.cpp  
BmpReg.icpp  
In file included from  
/home/patrick/Documents/OutProjsUpp/Bazaar/Docking/GCC.Debug.Debug\_Full.Gui.Shared/\$blit  
z.cpp:74:0:

scope

scope

Docking: 7 file(s) built in (0:00.17), 24 msecs / file, duration = 3631 msecs, parallelization 100%  
PdfDraw: 6 file(s) built in (0:00.24), 40 msecs / file, duration = 4112 msecs, parallelization 100%  
plugin/bmp: 4 file(s) built in (0:00.05), 13 msecs / file, duration = 3056 msecs, parallelization 100%  
Draw: 34 file(s) built in (0:02.38), 70 msecs / file, duration = 7418 msecs, parallelization 75%

There were errors. (0:08.02)

I am running on Ubuntu 12.10 and Upp Svn 6187.

Regards

Biobytess

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [mirek](#) on Sat, 13 Jul 2013 19:38:14 GMT

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

---

Even more weird. If you look to what is in repository, it seems to be OK:

[http://code.google.com/p/upp-mirror/source/browse/trunk/ups\\_rc/Docking/DockPane.cpp](http://code.google.com/p/upp-mirror/source/browse/trunk/ups_rc/Docking/DockPane.cpp)

(the method is commented out...)

Can you both please check whether you are using the 'correct' package? The one from uppsrc?

Mirek

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [Klugier](#) on Sat, 13 Jul 2013 20:12:42 GMT

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

---

Hello,

It seems that, there are two docking packages one from bazaar and one from uppsrc. Personally, I think that we should remove three following packages from bazaar: "Docking", "DockingExample1" and "DockingExample2". Two independent packages for docking contributes to unnecessary chaos. At the end, I would like to notice that "Docking" from uppsrc compiles, but "Dokcing" from bazaar doesn't.

Guys, Can you try compile packages from "reference" assembly (DockingExample1 & DockingExample2)?

Sincerely,  
Klugier

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [281264](#) on Sat, 13 Jul 2013 22:07:14 GMT

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

---

I've tested DockingExample1&2 in Reference and both work fine; the code lines in DockPane.cpp

update the corresponding versions in Bazaar.

Thx,

Javier

---

Subject: Re: Docking host platform dependency refactored

Posted by [mirek](#) on Sun, 14 Jul 2013 07:47:56 GMT

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

---

281264 wrote on Sat, 13 July 2013 18:07 I've tested DockingExample1&2 in Reference and both

(treated as comments); perhaps it would wise to update the corresponding versions in Bazaar.

Thx,

Javier

OK, removed from Bazaar...

Mirek

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [BioBytes](#) on Sun, 14 Jul 2013 12:34:25 GMT

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

---

Hello Mirek,

All is working perfectly when docking package is removed from bazaar keeping only the files in uppsrc directory.

Regards  
Biobytess

---

---

Subject: Re: Docking host platform dependency refactored

Posted by [Novo](#) on Sun, 14 Jul 2013 16:57:32 GMT

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

---

mirek wrote on Wed, 03 July 2013 01:47 Thanks for reporting. I am trying to fix all of that...

Anyway, are you sure that "tabify against floating" really worked in previous versions? I have checked the code (albeit not thoroughly yet) and I see no support yet...

Mirek

Sorry for the delay with the answer.

"tabify against floating" didn't work in previous version. I checked that against version from 2011-12-01.

One big difference between version from 2011 and current version is that floating window was a regular decorated window without a custom-made docking bar. Current floating window is

undecorated, but has a docking bar with no menu and a broken close button.

Another observation related to the docking bar menu. It is quite confusing. It uses combination of check box with sub-menu. This combination doesn't work as somebody would expect. In some cases actions caused by using this menu are just not logical. Example Auto-Hide --> Right and repeat the same Auto-Hide --> Right on the same control, you will end up with a docked control.

Another weird behavior: double-click on docking bar will undock window, double-clicking on floating window right after it became floating won't have any effect, but if you click on main window and after that you single-click on a bar of floating window it will dock back.

---

---

**Subject: Re: Docking host platform dependency refactored**

Posted by [281264](#) on Sun, 14 Jul 2013 18:37:27 GMT

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

---

Docking is an outstanding tool without question; however it has got some limitations in addition to what are mentioning. For example, if you use it with GICtrl in several windows/panes and assign and does not work well.

In my opinion, Mirek is the only person (I think James Thomas -the author- is no longer maintaining Docking) with enough knowledge to fix this aberrant behaviour.

By the way, thanks Mirek for adjusting and tweaking Docking.

Javier

---

---

**Subject: Re: Docking host platform dependency refactored**

Posted by [Novo](#) on Sun, 14 Jul 2013 19:31:33 GMT

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

---

I fill myself guilty for not making any code for U++. At least I do some QA work.

---

---

**Subject: Re: Docking host platform dependency refactored**

Posted by [koldo](#) on Wed, 17 Jul 2013 06:52:26 GMT

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

---

Novo wrote on Sun, 14 July 2013 21:31 I fill myself guilty for not making any code for U++. At least I do some QA work.

That is right!

---