

U++ - Bug #666

Provide access to underlying X11 Window of DHCtrl in GTK+ backend

01/26/2014 06:20 PM - Massimo Del Fedele

Status:	Rejected	Start date:	01/26/2014
Priority:	High	Due date:	
Assignee:	Miroslav Fidler	% Done:	0%
Category:	CtrlCore	Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description			
<p>Since introduction of GTK backend, the DHCtrl control (in GTK) is simply a Ctrl alias.</p> <p>I'd need access to underlying X11 window object in order to embed an OpenCascade window there.</p> <p>Now it's only possible in windows and in X11 backends, not in GTK one.</p> <p>The best would be to add a protected function GetHWND() to Ctrl, returning Null for non-windowes controls and the window handle for windowed ones, and make it public in DHCtrl.</p>			

History

#1 - 01/27/2014 09:12 AM - Miroslav Fidler

- Status changed from New to Ready for QA

- Assignee changed from Miroslav Fidler to Massimo Del Fedele

I am somewhat reluctant to provide this: move to GTK was primarily intended to isolate us from X11, so that when e.g. wayland takes over, we are safe. Or perhaps to support more platforms.

I think you should rather use

```
class Ctrl {
public:
    GdkWindow *gdk() const { return top ? top->window->window : NULL; }
    GtkWindow *gtk() const { return top ? (GtkWindow *)top->window : NULL; }
}
```

and then use gdk/gtk functions to retrieve X11 handle (possible with `#ifdef GDK_WINDOWING_X11`).

#2 - 01/27/2014 01:59 PM - Massimo Del Fedele

- Assignee changed from Massimo Del Fedele to Miroslav Fidler

Any solution for me is good, as long as I can get the platform window handle for a DHCtrl.

I'm creating an OpenCascade control and it needs a windowed control and its handle.

IMHO, the best would be to provide the handle from inside DHCtrl with a protected function, so when you adds new platform you just have to provide this function for new platforms and the rest stays almost platform-independent, but any way will do the trick for me.

If you add it in some way, I'll adapt my code.

#3 - 01/27/2014 03:00 PM - Miroslav Fidler

- *Status changed from Ready for QA to Rejected*
- *Assignee changed from Miroslav Fidler to Massimo Del Fedele*

Well, what I wanted to say is that for GTK backend the apparent handle is not X11 id, but GtkWidget * (or GdkWindow *)....

I am not quite sure whether creating something like

```
void *GetWindowHandle();
```

is a good idea - I think you would still have to provide platform specific code (plus it is not quite defined which one of 3 possible handles it should return for GTK/X11).

#4 - 01/30/2014 01:24 AM - Massimo Del Fedele

- *Assignee changed from Massimo Del Fedele to Miroslav Fidler*

Hi,

I tested with gtk() and gdk() functions, but they return NULL.

I guess (just lurked a bit inside your code...) because DHCtrl, in GTK mode, is simply an alias of Ctrl which don't get a window (aka top==NULL).

I think that DHCtrl should be a windowed control as in win32 and x11 mode, otherwise it becomes useless.

And I think, once it's windowed, it could contain a member function to get its handle.

The latter isn't mandatory, of course, but why have a DHCtrl then, if there's no access need to platform-dependent window ?