
Subject: Conceptual Query: widget functions access
Posted by [281264](#) on Sat, 17 Jul 2010 21:46:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

the TopWindow of an application.

toolbar or any other widget.

TopWindow?

Thanks,

Javier

Subject: Re: Conceptual Query: widget functions access
Posted by [mrjt](#) on Sun, 18 Jul 2010 07:54:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Two methods immediately suggest themselves:

- Overload ChildMouseEvent on the TopWindow to pick up MouseEnter and MouseLeave events on it's children. You can then use OverrideCursor to set the mouse cursor.

- Use a template:

```
template <class T>
class WithCursorImage : public T
{
    Image cursorimage;
public:
    WithCursor() { cursorimage = Image::Arrow(); }
    T& SetCursorImage(Image img) { cursorimage = img; return *this; }
    virtual void CursorImage(Point p) {
        Image img = T::CursorImage(p);
        if (img == Image::Arrow())
            img = cursorimage;
        return img;
    }
};
```

(from memory, so there may be some small errors though the logic is correct)

Subject: Re: Conceptual Query: widget functions access

Posted by [281264](#) on Sun, 18 Jul 2010 08:36:27 GMT

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

Thanks,

I shall test it.

Another question: How images can be used in ctrls if those controls are defined in separated .h and .cpp files (i.e. if they custom controls)? How to make .iml images available?

Best wishes,

Javier

Subject: Re: Conceptual Query: widget functions access

Posted by [281264](#) on Sun, 18 Jul 2010 11:03:02 GMT

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

I have made a version that works. What do you think? Is it "professional"? Is it well "built"?

Please your advise.

Thanks,

Javier.

File Attachments

1) [dummy_custom_ctrl.7z](#), downloaded 205 times

Subject: Re: Conceptual Query: widget functions access

Posted by [mrjt](#) on Mon, 19 Jul 2010 11:33:21 GMT

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

There are a few things (besides coding style):

```
virtual void MouseMove (Point p, dword d){  
pos_x=p.x;  
pos_y=p.y;  
Refresh();  
}
```

```
virtual void Paint(Draw& w){  
w.DrawRect(GetSize(), White());  
w.DrawText(10, 10, AsString(pos_x));
```

```
    w.DrawText(10, 20, AsString(pos_y));
}
This can be replaced by the more concise:
virtual void Paint(Draw& w){
    w.DrawRect(GetSize(), White());
    w.DrawText(10, 10, AsString(GetMouseViewPos()));
}
```

this code contains a bug:

```
virtual Image CursorImage (Point p, dword d){
    Image img=T::CursorImage(p,d);
    if(img=Image::Arrow()) <-- Should be ==
        img=Images::linguist_prev();
    return img;
}
```

and you can remove cursorimage and SetCursorImage, since you don't use them.

To use .iml files in header and source you include iml_header.h and iml_source.h in the .cpp file with the same IMAGE* defines.
