
Subject: Can a ctrl remove itself?

Posted by [Giorgio](#) on Mon, 16 Jul 2018 09:54:39 GMT

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

Is there a way for a ctrl to remove itself?

Subject: Re: Can a ctrl remove itself?

Posted by [Oblivion](#) on Mon, 16 Jul 2018 10:11:16 GMT

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

Hello Giorgio,

Quote: Is there a way for a ctrl to remove itself?

Yes, you can have a ctrl remove itself, by calling `Ctrl::Remove()`.

E.g.

```
class Foo : public TopWindow {
```

```
    Button bt;
```

```
public:
```

```
    Foo() {
```

```
        SetRect(0,0, 640, 480);
```

```
        CenterScreen();
```

```
        bt.SetLabel("Press to remove me!") << [=] { bt.Remove(); };
```

```
        Add(bt.HCenterPosZ(120).VCenterPosZ(24));
```

```
    }
```

```
};
```

This works, because:

Quote:

Ctrl Tree

Ultimate++ uses a linked list for all the child Ctrl's that have been Add()ed to it, partaking of its drawing space. The Ctrl does NOT own its children, but simply references them (`Ptr<Ctrl>`). They should be owned by your application, somewhere in a U++ container, i.e. `Array<Label>` or they are already made members of your application when using Layout files. If a Ctrl is added to another, it is ensured to be properly removed from its previous parent, thus a Ctrl cant be part of 2 trees.

[https://www.ultimatepp.org/srcdoc\\$CtrlCore\\$CtrlDesignConcept s_en-us.html](https://www.ultimatepp.org/srcdoc$CtrlCore$CtrlDesignConcept s_en-us.html)

Best regards,
Oblivion

Subject: Re: Can a ctrl remove itself?
Posted by [Giorgio](#) on Mon, 16 Jul 2018 13:50:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks!

Subject: Re: Can a ctrl remove itself?
Posted by [koldo](#) on Thu, 26 Jul 2018 08:32:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Good answer :)
