
Subject: Re: Proposed change to U++ to allow owning children.

Posted by [Lance](#) on Fri, 18 Mar 2011 14:21:01 GMT

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

kohait00 wrote on Fri, 18 March 2011 11:01 the flag handling should be better IMHO.

flag = 2 is quite bad, leaving things untouched...

it should determine itself, when the control is owned by someone else, to take over the responsibility

Sorry I misinterpreted you. Your point is valid.

Yes it's possible to revise flag from tribool to bool for all the AddChild*(and Add) functions. The implementation of the function will need to be changed accordingly.

Another option is to promote SetOwned(bool owned=false) from protected to public, and keep AddChild prototypes untouched.

```
...
public:
...
    // query owned flag
    bool IsOwned()const{ return owned; }
    Ctrl& Owned(bool b=false){ owned=b; return *this; }

...
};
```

And the way it's used will be changed to something like this

```
class App: public TopWindow{
...

App()
{
    Button * p = new Button();
    Button * p2 = new Button();

    this->Add(p.Owned());

    this->Add(p2); // Error, p2 is not owned.

    // fixed p2;
    p2.Owned();
}
```

```
};
```

Anyway, the detail can be polished and fixed to satisfy majority of users as long as there is no big holes in the design.
