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

Posted by [Lance](#) on Fri, 18 Mar 2011 13:39:38 GMT

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

Hi Kohait00:

Internally, the owned flag has but 2 possible values: true or false. When a Ctrl is removed from its parent by RemoveChild family of functions, there are 3 scenarios:

1. it's owned and should be automatically deleted by the parent control thereby it's destructor is called and the memory in the heap to house the Ctrl is also freed;
2. it's not owned. Basically this is the behaviour before the owned flag is introduced. So business as usual. Existing code should not feel any difference. They don't know anything about the owned flag, don't rely on it and will not be bothered by it.
3. it's owned and should be leave alone. Technically, it should be named DetachChild. But again, we don't want to any unexpected impact to users who don't know the existence of this newly installed capability. So I add an additionaly parameter detachOnly[to the RemoveChild family of functions, with carefully chosen default value so as no to break existing code. As you can see, in the AddChild function implementation I highlightly in a previous post, RemoveChild is called with detachOnly set to true so that it's not destroyed as our realy intention is just to change a dad for it.

Also, it could be useful to library user. User programs can detach a child and take appropriate actions. The only thing to remember is once a child is detached, the one who detach it has its ownership. It can either free it, or entrust it to someone else. The principle of "Everything belongs somewhere" is consistently confirmed.
