Subject: Re: revised ownership change
Posted by Lance on Thu, 28 Apr 2011 21:18:09 GMT
View Forum Message <> Reply to Message

Personally I don't dislike Detach at all. It's hard to think of a situation where Detach causes problem more than Array::Detach does. The only problem is that it will make the code harder to understand for somebody who doesn't go deep enough on this topic.

And if a Child control can be Detached, could it be reattached to another parent? If the answer is not, than Detach would be perfectly fine. If the answer is yes, essentially you open the door for add a user newed object which Mirek objects.

So either:

1. Call parent.CreateOwned<ChildType>() to create the object on heap and add as parent's child;
2. When the child is expected to be killed, hide it, access all its properties, and call parent.Remove(..)

or

1. Call parent.CreateOwned<ChildType>() to create the object on heap and add as parent's child;
2. When the child is no longer needed, Detach it and remember the returned poiter.
3. Access its properties, and delete it manually.

The advantage of case 2 over case 1 is that it allow the parent be destructed while the child remain valid, think about the case where the parent itself is its parent's dynamically created child.

The disadvantage of case 2 over case 1 is that it requires the library user to use delete. But since he/she has to delete the object returned by Array::Detach, it should not hurt that much.

If you agree with my above analysis, I go with case 2 too. Or do you have other recommendation?