

Hi all,
Windows XP 32 + SP3; MSC9; U++ 2467.

Using TabCtrl with an array of same layout I have this problem:

```
//main.h
```

```
Array<WithIDevice<ParentCtrl> > layer;  
TabCtrl          tab;
```

```
//main.cpp
```

```
void App::DoSomething_1(void)  
{  
    for(int i=0; i<5; i++) {  
        CtrlLayout(layer.At(i));  
        tab.Add(layer.At(i).SizePos(), Applmg::....(), t_("..."));  
    }  
}
```

```
void App::DoSomething_2(void)  
{  
    for(int z = tab.GetCount() - 1; z > 0; z--) tab.Remove(z);  
    tab.ClearFrames();  
    tab.GetItem(0).SetImage(Nullr()).Text("");  
    layer.Clear();  
}
```

First step:

Calling DoSomething_1() the application will add 5 tabs and all work fine.

Second step:

Calling DoSomething_2() the application will remove 4 of 5 tabs. (Using Reset() result in a system crash, the system crash also if trying to remove alla the 5 tabs).

Third step:

Calling DoSomething_1() again result in a system crash at the line: tab.Add(...);

Questions:

- 1) It is correct that tab.Reset() result in a system crash? I must leave one tab ?
- 2) Adding new tabs after remove is not possible?

3) Can be windows problem? or compiler?

Note that the posted code it's only an abstract.
Thanks a lot.

Subject: Re: Question and problems around TabCtrl
Posted by [mirek](#) on Fri, 23 Jul 2010 09:24:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

mubeta wrote on Thu, 22 July 2010 16:12Hi all,
Windows XP 32 + SP3; MSC9; U++ 2467.

Using TabCtrl with an array of same layout I have this problem:

```
//main.h
```

```
Array<WithIDevice<ParentCtrl> > layer;  
TabCtrl          tab;
```

```
//main.cpp
```

```
void App::DoSomething_1(void)  
{  
    for(int i=0; i<5; i++) {  
        CtrlLayout(layer.At(i));  
        tab.Add(layer.At(i).SizePos(), Applmg::....(), t_("..."));  
    }  
}
```

```
void App::DoSomething_2(void)  
{  
    for(int z = tab.GetCount() - 1; z > 0; z--) tab.Remove(z);  
    tab.ClearFrames();  
    tab.GetItem(0).SetImage(Nullr()).Text("");  
    layer.Clear();  
}
```

First step:

Calling DoSomething_1() the application will add 5 tabs and all work fine.

Second step:

Calling DoSomething_2() the application will remove 4 of 5 tabs. (Using Reset() result in a system crash, the system crash also if trying to remove alla the 5 tabs).

Third step:

Calling DoSomething_1() again result in a system crash at the line: tab.Add(...);

Questions:

- 1) It is correct that tab.Reset() result in a system crash? I must leave one tab ?
- 2) Adding new tabs after remove is not possible?

3) Can be windows problem? or compiler?

Note that the posted code it's only an abstract.

Thanks a lot.

I might be wrong, but it looks like in DoSomething_2 you do layer.Clear(); while layer[0] is still part of TabCtrl. I believe this is the cause of troubles.

Subject: Re: Question and problems around TabCtrl
Posted by [mubeta](#) on Fri, 23 Jul 2010 09:54:35 GMT

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

I found the cause, but I need a long time for follow all the codes:

So:

After add an tab, I also set a callback to WhenSet:

```
tab.WhenSet << THISBACK(foo);
```

foo get information from tab, so, when tab.Reset() is destroying the last one, (tab.Reset() method call again WhenSet() before to return), I get the crash.

WhenSet() is called again, at the second time adding loop, in this case it's really dangerous, because all the need vars are not ready.

So, in fact:

First step: adding tabs all is ok, because WhenSet is empty.

Second step: assign a callback to WhenSet.

Third step: resetting the tab result in a crash because Reset() also call WhenSet(), and in MY CASE is not good.

Third step: adding new tabs result in a crash because WhenSet is assigned but my vars are not ready.

So, now the problem is: how to remove the WhenSet callback??

```
tab.WhenSet << 0; (compiler error)
```

```
tab.WhneSet = 0; (compiler error)
```

```
tab.WhenSet << CNULL; (compiler error)
```

```
tab.WhenSet << Callback::Empty(); (ok for compiler, but nothing change in the application. the callback in fact is not cleared).
```

uff.

I only need to be able to remove the WhenSet callback some times.
