Subject: Re: Problem in example code
Posted by dolik.rce on Sun, 03 Jan 2016 20:08:28 GMT
View Forum Message <> Reply to Message

> vegaonline wrote on Sun, 03 January 2016 14:41
> However, I may reach Plot menu from any other menu item having separate layout except card menu. Should I use ".Remove() for all other different layout or there is any single Remove command to remove all other layout which may be present there?

There are ways to access the all the children of given Ctrls and to iterate through them. See GetFirstChild(), GetNext(), GetChildIndex(), GetIndexChild(), GetChildCount() and related functions in Ctrl documentation. In your particular case it would probably be ok just to call GetLasChild()->Remove(). But I never actually needed to use such a low level approach in my apps and it feels kind of error prone to me.

There is actually much simpler way: You can just remember your last state :) For example, you can add Ctrl* active to your class (don't forget to initialize it to NULL in constructor, otherwise you'll get crashes ;)), and then use a method like this for the switching:void

```
Testing::SetLayout(Ctrl* ctrl) {
 if (active)
  active->Remove();
 Add(*ctrl);
 active = ctrl;
```

}Your menu code could then be as simple as this: menu.Add("Configure",

```
THISBACK1(SetLayout, &doCard)).Help("Edit Simulation Setup");
  menu.Add("Geometry", THISBACK1(SetLayout, &doPlot)).Help("Edit Experimental Geometry");
```

That is of course assuming there is no special initialization for each layout needed on each switch, in which case you would have to still call doConfig() and doGeom() from menu and they would in turn call the common switching code in SetLayout.

Honza