

hi all..

here is an excerpt of Ctrl:: functions that need any doc comment. some are very selfexplanatory but some, like Update() or PostInput() aren't. extenders will appreciate that grain of info..

```
virtual void PostInput();  
virtual void ChildFrameMouseEvent(Ctrl *child, int event, Point p, int zdelta, dword keyflags);  
virtual Size GetMaxSize() const;  
virtual void Updated();  
virtual String GetDesc() const;
```

```
Callback WhenAction;  
bool IsForeground() const;  
void RefreshLayoutDeep() { SyncLayout(2); }  
void UpdateLayout() { SyncLayout(); }  
void UpdateParentLayout() { if(parent) parent->UpdateLayout(); }  
Rect GetWorkArea() const;  
bool IsIgnoreMouse() const { return ignoremouse; }  
Point GetMouseViewPos() const;
```

...

i'd add some but don't understand some of them myself.
Can anyone shed some info on it?

i.e.

Update(): Should be called whenever internal state of a Ctrl was changed, will invoke SetModified() and Updated(), so derive classes can adjust their additional state / computation info to it.

Updated(): see Update(). override it to add custom behaviour on internal state change of underlying base class (Ctrl or derived) before calling base class Updated(). Can serve to write back to data bases or update any back end representation..

is that true? how is the proper use case for that one, and what about PostInput() and GetMaxSize()?

cheers and thanks

EDIT:

BTW: why not make `Ctrl::Action()` virtual as well? that would enable us to easy extend action function without connecting to `WhenAction` callback, which would remain reserved for the user space??
