Subject: Re: Sharing and Locking
Posted by mirek on Tue, 16 Mar 2010 22:50:15 GMT
View Forum Message <> Reply to Message

gridem wrote on Tue, 16 March 2010 03:04luzr wrote on Tue, 16 March 2010 08:02
I would stop right there and asked "why?" I would prefer using the data instead of pointer.

The answer is simple: U++ already uses the same idiom . See for example:

CtrlCore.h:

```
static  Ptr<Ctrl> focusCtrl;
static  Ptr<Ctrl> focusCtrlWnd;
static  Ptr<Ctrl> lastActiveWnd;
static  Ptr<Ctrl> caretCtrl;
```

But these are to solve hard to predict user inputs. In most cases where MT threads are involved, you have much better control than that.

Quote:
No, the considered situation is a bit more complicated. Because I used not shared_ptr for global variable but weak_ptr, the object will live until it will be destroyed in thread 1. But if I was successfull on converting from weak_ptr to shared_ptr, than the object lifetime will be longer and will be destroyed when loop in thread 1 and thread 2 will be restarted. In any case the object will not be in partial (or zombie) state when it will be destoyed in destructor instead of some method like Close, Destroy or other.

Well, that is not what I mean. What is bad about shared ownership is exactly that it makes the lifetime of object unpredictable.

Quote:
The problems may occurs when I want to create the real MT application without GUI and try to access to global variables or global list of variables through Ptr.

Which is something to avoid, I agree...

Mirek