Subject: Re: Sharing and Locking
Posted by gridem on Fri, 19 Mar 2010 06:44:40 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 17 March 2010 01:50Quote:
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.

Yes, lifetime object will be unpredictable in sense that if my shared_ptr will be destroyes that the object itself cannot be. But it's not a problem in most cases, you can treat it as automatic garbarge collector for C++. So if clients (thread 2) want to use object than you (thread 1) should not prevent them from any operation even if you don't need it. In my practice I cannot remember the situation when an unpredicted lifetime would be a problem.