Subject: Re: Sharing and Locking
Posted by mirek on Mon, 08 Mar 2010 00:42:08 GMT
View Forum Message <> Reply to Message

gridem wrote on Sun, 07 March 2010 10:38Mirek, thank you for your answer.
luzr wrote on Sun, 07 March 2010 16:40
1) Ptr and Pte are VERY different beasts as compared to shared_ptr (but you probably know that).

Yes, it different but tries to solve the same kind of problems IMO.

Well, you can say that, but it is a bit far-stretched IMO. Pte/Ptr are solely for solving dangling pointer issue. Unlike shared_ptr (correct me if I am wrong), Ptr can point to stack objects and most of time they really do.

Quote:
I think that shared_ptr has more cleared semantics than Pte/Ptr. May be the reason of this is that I used shared_ptr a lot before and try to use Pte/Ptr like shared_ptr.

I wonder how you can even do that?

Quote:
So I mean that Pte uses Mutex (more precisely, StaticMutex) and in general it can have some problems in high concurency application. Lock-free implementation like atomic operations produces better performance results in general.

Yes, this correct, Pte/Ptr is not great perfomance-wise. (OTOH, Mutex is just two atomic operations