Subject: Re: Use same variable in different threads
Posted by dolik.rce on Thu, 09 Dec 2010 17:57:33 GMT
View Forum Message <> Reply to Message

Hi Koldo,

From my little knowledge (close to yours probably ):

Atomic is just typedefed int, which can be manipulated by U++ functions Atomic{Read,Write,Inc,Dec,XAdd}(). They are implemented using atomic operations, so it should be guaranteed that the return value is correct even if other thread changes value of the variable while processing the function call.

Volatile tells compiler that the variable can be changed from other threads, so that compiler knows that the generated code must check for its value everytime, instead using for example value stored in cpu cache, because other thread might have changed since it was stored there.

Mutex is a mechanism that allow you to lock some code so that if other code gets to the same section it has to wait for the first one to leave. This is utilized by INTERLOCKED macro in U++.

RWMutex is a variant of mutex that allows you to mark "read code" which can be executed by any number of threads at once and "write code" that can be executed only by one thread at a time and no thread can run in write mode until all reading threads are finished.

Hope that at least some of it helps... It definitelly helped me to sort it out in my head

Best regards,
Honza