Subject: Re: Job package: A lightweight multithreading tool, using promise/future mechanism
Posted by Oblivion on Sun, 10 Sep 2017 22:25:12 GMT
View Forum Message <> Reply to Message

Quote:

That one is a borderline to impossible.

The problem are all those destructors that need to called. It is hard to abort thread that is does not know about it...

Ah no, what I mean is setting a unique flag for each thread (maybe using ThreadId), so that it can be checked from within like IsShutdownThreads.
(I do this with Job:Cancel() method and IsJobCancelled() global function, by keeping an index of their IDs, which are integer numbers, incremented with each new thread, using a thread local variable as the unique id.)
I was just talking about signalling the shutdown to specific targets so that user can design his/her MT code and handle its shutdown conditions easily.

And this is exactly where Job comes handy. It is a higher level interface, a convenience wrapper, if yow will, with a simpler return semantics, error management, and a more refined shutdown mechanism. With a reasonably small overhead.

Best regards,
Oblivion