Subject: what about WorkQueue : public CoWork
Posted by kohait00 on Wed, 03 Feb 2010 06:26:08 GMT
View Forum Message <> Reply to Message

hi there..

recently got an idea..

there is type of work which should be done somewhere in a worker thread, but in given order. something like "run it somewhere, *not* in this thread, i dont care where, but keep the supplied order of work". to be sure that certain things dont get executed before other things, but they *do* execute somewhere in parallell to current thred. Does that make sense?

more or less the approach of PostCallback, but there is one decent thread running all the work sequentially. something like:

WorkQueue : public CoWork /*with only one thread*/

WorkQueue wq;

/*in any thread, main thread probably, offload work */

wq & THISBACK(A)
wq & THISBACK(B)
/*any other work*/
wq & THISBACK(C)

/*the single worker thread */
DoJob(A)
DoJob(B)
/*maybe done, sleeping */
DoJob(C)
...

hope you got the point. it's almost like CoWork, but suppying to CoWork one needs to have real paralellable work, one cant be sure of what is going to be executed first, what later.. depends..or at least it is really executed in parallel. (or am i wrong)

so deriving from CoWork would almost do it, if ensuring that only 1 thread is the worker thread there. (I know that CoWork uses a static Pool of threads. that needed to be tweaked)

benefit: one could set up a decent count of worker threads, that could be responsible for different type of work, that can be executed in parallell, but the work that is "dependant" would still execute sequentially. this would make life esier in some cases, not needing to provide a lot of IPC anymore. just by logical *pre* runtime separation of work.

comments please