Subject: Re: [PROPOSAL]: AsyncQueue class (a single threaded synchronization tool) for U++
Posted by mirek on Mon, 29 Feb 2016 16:23:46 GMT
View Forum Message <> Reply to Message

Just a small nitpicking for now...

```
 for(int i  = 0; i < test_stack.GetCount(); i++) {
.......
   test_stack.Remove(i);
```

this combo is generally incorrect - it skips the element after the removed one. It probably works OK here (the next element gets tested/removed in the pass), but still...

if(WhenDo) WhenDo()

No need for if, it is OK to call unassigned Callback - it is NOP.

```
   template<class T>T& GetJobArg(int i)                    { return job_queue[i].Get<T>(i); }
```

I smell the bug here - is 'i' index of VectorMap or index of VarArgs?

Overall, I am pretty unsure what is that VectorMap useful for. You are never using it as map,except for RemoveJob. Not that all those VectorMap::Insert/Remove operations are quite slow.

In practice, I am not quite sure you really need a queue there. Do we really need more than single level?

Also, we are going full C++11 soon. Lambdas would make superior alternative for both DoJob and VarArgs IMO.

Mirek