Subject: Re: Thread based "conveyor" class for Win32
Posted by mirek on Wed, 10 Oct 2007 17:35:36 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Wed, 10 October 2007 12:51In my work I have rather usual task of having some queue service, processing main thread requests in background. I just finished general purpose class template for simplified handling of such a task. It`s rather young (with interface unpolished) but look at what you can do with it:

```
template<class T> class ConveyorThread : protected Thread
{ typedef ConveyorThread CLASSNAME;
public:
 ConveyorThread(bool enabled = true);
 virtual ~ConveyorThread();
 void Enable(bool enable = true);

 void Request(Callback1<const T &>handler, const T &request);
 void ClearAllRequests();

 void SharedEnter();
 void SharedLeave();

 void RequestFinish();
 int  WaitFinished();
 bool IsFinished();
}
```

Just add request for asyncronous processing and define handler for it - and that`s all. Rather convenient, I suppose.
Class also has guarding functions for interacting with main thread`s objects: SharedEnter/SharedLeave (for updating interface, statistics, etc).

Any suggestions, critics and recommendations is appreciated.

Interesting idea. Is there anything that prevents using Semaphore instead of events there? (The problem is that your code is win32 specific).