Subject: Re: MT & File Access
Posted by mirek on Thu, 08 Mar 2007 13:08:25 GMT
View Forum Message <> Reply to Message

victorb wrote on Mon, 05 March 2007 19:23My application is a kind of viewer from files downloaded from the Web.

The GUI thread is the viewer.

There is also an updater thread that check for updates.

Im my current implementation, the GUI thread read local files written by the updater thread. Reads (GUI) and Writes (updater) are serialized.

I am not sure if this approach is OK:
- the GUI could be writting while the updater is also writting (i.e. Debug log),
- the GUI might be using FindFile while the updater is writting some files.

Is my implmentation right or should I gather all file acceses in a single thread ?

Any help is welcome (I'm new to MT),
Thanks,
Victor

Well, generally, what you want to avoid is race condition, which almost always is caused when read(s) and write(s) to a single entity (file, variable) occurs at the same time.

In this case, situation is a little bit more complicated, as in fact written files are "external" entities; therefore you will most likely have to use some sort of file protection, similar to the one used to avoid simultaneous access from several processes.

Alternatively, you migh create some internal locking mechanism. I am afraid the topic and possible solutions are too widespread to give a better advice.

Well, maybe what you want to hear is: Yes, you can write one file on one thread and another file in another, that is not problem.

Mirek