
Subject: Re: Some projects are ready to become open-source

Posted by [Mindtraveller](#) on Tue, 06 May 2014 22:38:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

PersistentMap - a bunch of templates to make U++ vectors/maps persistent.

The idea behind it is very simple:

- 1) Take original storage class
- 2) Restrict or don't restrict access to it storage itself depending on flavour.
- 3) Add functions for "mt safety".
- 4) Add background thread to write data to disk if storage is changed with some timeout (if it is changed frequently).
- 5) Make writes atomic and stable across situation of write failure / io failure / power failure, so that user code transparently works with stored data even after failure.

currently added is

- 6) Versioning support for "open" flavour

and to be done is

- 7) Clustered incremental disk flushes for partial and effective disk writes.

First flavour is "restricted" PersistentStorage/PersistentMap<STORAGE>. This template doesn't give access to underlying container. Instead it introduces functions like Add, AddPick, Get, GetFirst, Put, CheckAdd or GetRemoveFirst. You may be sure about it's "mt safety" and persistency.

But this is not always enough. Sometimes you need more flexibility, and here you have "open" flavour, a OpenPersistentStorage<STORAGE> class. It gives you full access to underlying storage, but you are responsible for calling EnterRead()/LeaveRead() and EnterWrite()/LeaveWriteSetChanged(bool changed).

Currently these templates are in release stage and you may use them without any doubt of losing your data. Any suggestions or proposals are welcome.

File Attachments

- 1) [PersistentMap.h](#), downloaded 499 times
-