

---

Subject: Re: Crazy(?) idea about debugging  
Posted by [mr\\_ped](#) on Wed, 18 Mar 2009 15:01:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Interesting.

But things to keep in mind:

- a good old full memory dump would lead to the cause of exception in ordinary case faster, at least the exact exception instruction is easier to fetch, the true cause may be more hidden. (and it may be lot more hidden than history of 100 values anyway)
- with non-atomic variables you may get half way updated values in history, it would be next to impossible to keep those data consistent unless you make cooperative call to the history keeper to let him know when it's safe to store history (at that point the whole concept becomes too cumbersome, you can do that on your own with your own data history watcher and calling it periodically).
- lot of C/C++ code works with local variables at heap, they will be very difficult to follow+store in the history manager.

I \*believe\* unit testing with QA level tests (i.e. not just 100% code coverage for TDD, but additional quality tests with wider range of data) is better and cheaper safety net, although it's far from universal solution, because some problems are difficult to unit test. Also QA level of tests gets into the way of TDD, so it makes sense to use just minimal set of tests to do TDD and after the module does reach some "1.0" version, put it under QA tests with additional range of data. After that minor refactoring should change all the tests (a bit cumbersome, but usually tests don't change as much as code, because usually you want to keep the output same, so it's workable), or in case of huge refactoring it's probably better to drop QA part, work with just original minimal set, and then introduce new QA tests for "2.0".

This should bring the need of such idea to minimum, because such well tested code usually doesn't crash at all, or does crash from such complex conditions and hidden faults, that data history with 100 last values may be completely useless to figure out where it went wrong.

... sorry if I sound too negative, maybe I need more time to let the idea sink down to appreciate it more.

---