
Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by [mr_ped](#) on Mon, 20 Dec 2010 14:52:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

The reasoning behind assert vs throw is performance.

For exception throw you need a test, that means every Vector item fetch to test whether index is valid. That would be quite a huge performance penalty, so Vector is designed in "insecure" way, and it's up to you to call it with only valid indexes.

The assert does add those tests, but only in debug mode, and there's no point to continue in the application further, because in production mode it would crash anyway, so if you hit assert in debug, you should fix your code to not get into the state which triggered assert, never ever.

It's quite a C++ way, but it's more about deciding which part of code is responsible for data validation, and which part is already trusted and simply does what it should without additional checks. So it's not so much language dependent.

In interpreted languages like python there's usually every instruction validating everything, because it's anyway interpreted+running in sandbox and the VM has to catch any such problem anyway because of security reasons. After that it's very cheap to throw it up as an exception, because the detection mechanism is (and must be) already there.

In C++ the wrong code will simply crash the machine. In modern OS the process is isolated enough that it will not hit other processes, but that's thanks to the OS protection, the C++ code as is doesn't care at all. If you want any checks, you have to write them, otherwise you are running at full speed without any hidden automagic code supplementing your work of coder.
