Subject: Re: Raise Exception instead of ASSERT for container
Posted by mirek on Tue, 09 Oct 2007 21:33:45 GMT
View Forum Message <> Reply to Message

benoitc wrote on Tue, 09 October 2007 12:26The goal of exceptions in C++ is to clearly separate nominal code with exceptional errors management, instead of adding a bunch of test before calling any API just in case an error occurs.

Mostly correct, but "errors" is confusing. Exceptional situation is much better term. Such exceptional situation are intiated by conditions that cannot be predicted by program logic, like not enough space on HD or invalid format of file being read.

Quote:
Without exception you need to check first that "i" in inside the bound then get a reference to array[i] and then check than "j" is inside the bound of the array[i] container.

Incorrect (IMO). The contract of Vector::operator[] has precondition that index is in 0 - GetCount() range.

Quote:
Don't you think that it might be easier to use exception to handle potential error on array access?

Definitely not. Exception are not there to fix programmers errors. At least not in U++ and I believe not even Bjarne/Stepanov/Koenig et al meant such thing to be.

Well, something completely different is if the index would be result of some input, e.g. it woul be in input file. Then using exception to stop parsing of intput file is completely valid.

However, that should be done by parser code. If nothing else, testing the index in each operator[] would make code in some cases as much as 500% slower (as usually compiler is able to optimize the loop to the pointer scan).

Quote:
The usage of exception is more a matter of taste than anything else, so if you're not convince that it might be useful for that kind of stuff, I'm fine with that.

Well, do not get me wrong, I think this is a useful debate.

Plus, you still have not explained how exceptions are going to help with multidimensional arrays....

(BTW, v.At(x).At(y) is method to deal with them in U++).

Mirek