Subject: Re: Raise Exception instead of ASSERT for container
Posted by benoitc on Wed, 10 Oct 2007 08:46:24 GMT
View Forum Message <> Reply to Message

OK, It was just a suggestion, I think we clearly don't have the same perception of the exception usage in C++:

Quote: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: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.

I'm not that sure about that, because if you refer to C++ bible (Bjarne) in the chapter 14, paragraph 14.1.1 (Alternative Views on Exceptions), quoting the Master
Quote:Bjarne:"Exception is one of those words that means different things to different people"
Here at least we all agree
Quote:Bjarne:The C++ exception-handling mechanism is designed to support handling of errors and other exceptional condition(hence the name)... This mechanism is designed to handle only synchronous exceptions, such as array range checks...

It looks like I'm not that far from the definition of the Master. For me, as soon as you have a function that return directly a value/object/reference and that can potentially fail, you could/should throw an exception to let the user of that API deals with such case and avoiding adding a bunch of test before the call (sometime it is not even possible).

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

I thought I did: How to you check nicely that j is a valid index in the v[i][j]? I personally think it is a pain to do if(i>=0&&i<v.GetCount()&& j>=0 && j<v[i].GetCount()) (imagine in the case of 3 or 4 dimension) each time I want to access the array, I would prefer that to be handled by the vector class itself through exception.

BTW, I think that v.At(y) if a little bit different than
v[y], because in my case I don't want to increase the size, I want to access an already allocated an populated array, but I use a index (retrieve from a editable file) that can be wrong (<0 or >GetCount).

As I said, It was just a suggestion based on my perception of the exception usage in C++ or in Java. If you don't want to use that in U++, I cannot blame you, your code is so excellent than I'm fine if only one minor thing does not please me in U++.

Regards,
Benoit