Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by nlneilson on Tue, 21 Dec 2010 00:07:48 GMT
View Forum Message <> Reply to Message

Very good explanation, I appreciate that.
I am very new to Vector<String> and assert.

I have a header file to parse data that is more than 400 lines now which is basically the concept used for more than 10 years.
Instead of using split it handles each character:
    cc = ln.GetCount();
    for ( int i = 0; i < cc; i++ ){

In Python the test for loops is 1,000,000 times (if I remember right).  Reading a line to split it then to read the chunks is slower than reading each character and acting on it once.
I remember the first computer I had, a Radio Shack notebook that was extremely limited and used a cassette tape for extra memory.

So I just changed the code in the first post to:
    try{
// Vector<String> vBL = Split(BLoc, ',');
// ln = vBL[0] + "," + vBL[1];// post error to forum
 ln = parseLatLon(BLoc);
and that returns "Invalid data", the user is informed and all is OK.

I ran into this problem when porting Python/Java code to C++.
A very short app that synchronizes two GPS files and corrects for deviations because of atmospheric or other conditions.
I tried the Vector<String> Split() as a quick first try.
The base location can actually be in decimal degrees, degrees&minutes or degrees&minutes&seconds when run through the parse code.

I spend much more time trying to handle problems than getting something to work under controlled/ideal conditions.  When I input data I knew was wrong to see what the results were is when the app crashed in debug mode and MS killed it.

Previously I thought try/catch would catch all errors in C++ the same as in Python and Java.

Quote: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.

That is something my simple mind can comprehend, Thanks.

Neil