Subject: Fatal error kills app in try/catch - - Why?
Posted by nlneilson on Sat, 18 Dec 2010 08:32:05 GMT
View Forum Message <> Reply to Message

Fatal error
Assertion failed in c"\upp\uppsrc\core\Vcont.h,line 33
i>=0 && i<items

MS kills the app.

I know where the problem is:

```
   try{
 Vector<String> vBL = Split(BLoc, ',');
 ln = vBL[0] + "," + vBL[1];// post error to forum
 ln = parseLatLon(BLoc);
```

If a user inputs data that is space rather than comma delimited then vBL[1] is null.

In Util.cpp:

```
 __BREAK__;
 abort();
```

In Vcont.h:

```
 T&       Get(int i) const        { ASSERT(i >= 0 && i < items); return vector[i]; }
```

The parseLatLon(BLoc); code is also in a try/catch block.
The question I have is since this is in two try/catch blocks WHY is the error not caught rather than have MS kill the app??

Neil

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by cbpporter on Sat, 18 Dec 2010 14:31:16 GMT
View Forum Message <> Reply to Message

Because assert like mechanisms and exceptions are two different things. AFAIK U++ rarely uses exceptions.

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by nlneilson on Sat, 18 Dec 2010 21:07:10 GMT
View Forum Message <> Reply to Message

Thanks for the response. I had not run into "assert" enough to find out the purpose, I have been using the interpreted languages Java and Python before switching to C/C++ using Upp.

With a little research:
Asserts only kills the app in debug and not release.
Good for testing.

In Java and Python I used try/catch blocks extensively to handle errors. When the code was ported to C++ they are still there and seem to work OK except in certain cases like the one mentioned.

cbpporter wrote on Sat, 18 December 2010 15:31 AFAIK U++ rarely uses exceptions.

What does U++ use to catch errors if not "exceptions"?
I thought try/catch and exceptions were together.

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by cbpporter on Sun, 19 Dec 2010 10:17:32 GMT

View Forum Message <> Reply to Message

No, U++ uses try/catch to catch exceptions. The thing is, not everything trows exceptions like in Java and Python. See you example with the Vector. Only a few things throw exceptions, but there are plenty of asserts. XML and CParser are two things that I can think of from the top of my head that throw exceptions.

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by nlneilson on Sun, 19 Dec 2010 21:55:25 GMT

View Forum Message <> Reply to Message

Thanks. Will keep this in mind.
Being used to Java and Python there are habits there I need to get over when using Upp. I am new to many things in C/C++.

One other question on assert.
Is this more specific to Upp or C++ in general as far as "not everything trows exceptions"?

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by cbpporter on Mon, 20 Dec 2010 09:54:27 GMT

View Forum Message <> Reply to Message

Well I don't know. C++ has little to prevent use of exceptions but most libraries I saw don't use them. On the other hand, there is very little "real" C++ out there. Standard library and MFC does not use exceptions too much or at all if I'm not mistaken.

Qt does not either.

I never saw C++ throw exceptions when segfaulting, like Java, but would like to see it.

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.

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

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by nlneilson on Tue, 21 Dec 2010 02:54:57 GMT

View Forum Message <> Reply to Message

Another way this can be handled is:

```
    try{
 Vector<String> vBL = Split(BLoc, ',');
 if(vBL.GetCount() < 3){ BaseLoc<<= "Invalid data " + BLoc; return;}
```

In the BaseLoc EditField a user is informed the data is incorrect and the data input is still there.

Since space is an issue that is preferred rather than the big Exclamation() pop up. The third data field is Altitude which is used a few lines down in the code.

Searching the net found for Vector .size() that worked but the Upp pop up had the option of .GetCount.

What difference is there for .size or .GetCount?
edit: Apparently they are the same except .getCount was not introduced until VC++ 7.0.
http://www.codeguru.com/forum/showthread.php?t=333640

Now to search the MyApps directory for Vector and add this error handling where necessary, much easier than getting a bug report and fixing it later.

Neil

Subject: Re: Fatal error kills app in try/catch - - Why?
Posted by cbpporter on Tue, 21 Dec 2010 11:18:46 GMT

View Forum Message <> Reply to Message

I think you are confusing vectors. There is "vector" and "Vector". The first has size, the second GetCount. More precisely, they are std::vector and Upp::Vector. There is also CArray and your link only applies to CArray.