
Subject: Different results from same code&settings
Posted by [dolik.rce](#) on Sun, 07 Sep 2008 20:50:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
I'm working on a project in U++ for about two months with no problems whatsoever, but lately I've seen some weird things to happen:
I compile main package, execute the program and everything works as it is supposed to. Then I compile it again, without any changes in code and it crashes in the middle of execution with "Invalid memory access!" error. After that, I have to replace the files from my backup and restart the ide to make it work again.
I tried to find what exactly causes it, but with no luck... I reproduced this behaviour in version 2008.1 and also svn.324 (from .deb package), both under Linux.

Had anyone encountered something similar or have you any idea what could be the cause? I'm really confused...

Thanks, Honza

Subject: Re: Different results from same code&settings
Posted by [mr_ped](#) on Sun, 07 Sep 2008 20:59:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

"replace the files from my backup"
So are the files corrupted after such crashing run?

I don't see how this can be related, if you don't change anything in the sources and they remain intact.

What can make difference:

- first vs second compilation

especially if you restore your files from backup, they will have older timestamp, which will make BLITZ to behave more aggressively, compiling them very likely all in one big file.

Once you do some minor change (like space/enter) to some of the files, it will be not included into the big BLITZ for next X minutes, so the compilation process differs then.

- did you try "Clean [package]" or even "Rebuild All"?

Sometimes some older object files are not recompiled when they should, especially if you keep restoring your source files, thus causing havoc to files timestamps. Rebuild All is a very basic step in search for mystic crashes.

- do you try debug vs release mode? This can make big difference for uninitialized variables content and their misuse. Sometimes debug version works even with bug, release will crash.

- there can be plenty of reasons for crash, from compiler bug (gcc 4.2 is bad, are you at 4.1 or 4.3?) to bug in application to HW problem. Hard to tell without actual sources.

Try also Debug mode with "Run (in debugger)", maybe TheIDE will catch the crash and show you the place in source.

Subject: Re: Different results from same code&settings

Posted by [dolik.rce](#) on Sun, 07 Sep 2008 22:15:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks for suggestions. I did more testing and I think I found the reason of the problem... I checked the MD5 checksums of all files in package folder to those in backup - the only different was .upp file. After checking them in text editor I found that the only difference is in the order of files in file section.

I realized that I was playing with the order before (just to make them look organized). I tested if this is it by executing "rebuild all" command in release mode with different ordering of files... and it was really it.

Now I must admit that my project is not exactly "usual" use of theIDE. I'm using Bison&Flex through "Custom build steps" so that must be the cause - somewhat wrong order of parsing and compiling files...

Only thing that still puzzles me is that the report in console always shows the .y & .l files being parsed before compilation of their output .cpp's. Here is the output if it helps:

```
----- Core ( GCC SHARED LINUX ) ( 1 / 3 )
```

```
cd /media/hda6/upp/uppsrc/Core
```

```
----- plugin/pcrc ( GCC SHARED LINUX ) ( 2 / 3 )
```

```
cd /media/hda6/upp/uppsrc/plugin/pcrc
```

```
----- Fam_a3.3 ( MAIN GCC SHARED LINUX ) ( 3 / 3 )
```

```
cd /media/hda6/MyUppApps/Fam_a3.3
```

```
parser.y
```

```
bison++ -d --verbose -h parser.h -o parser.cpp parser.y
```

```
bison++ -d --verbose -h parser.h -o parser.cpp parser.y
```

```
Exitcode: 0
```

```
scanner.l
```

```
flex++ -d -o scanner.cpp scanner.l
```

```
flex++ -d -o scanner.cpp scanner.l
```

```
Exitcode: 0
```

```
Fam_a3.3.cpp
```

```
g++-4.1 -c -I"/media/hda6/MyUppApps" -I"/media/hda6/upp/uppsrc" -I"/usr/include/freetype2"
```

```
-I"/usr/include/gtk-2.0"
```

```
-I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include" -I"/usr/include/cairo"
```

```
-I"/u
```

```
sr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC -DflagSHARED -DflagLINUX
```

```
-DbmYEAR=2008 -DbmMO
```

```
NTH=9 -DbmDAY=7 -DbmHOUR=23 -DbmMINUTE=47 -DbmSECOND=40 -fexceptions -Os
```

```
-finline-limit=20 -x c++ "/media/hda6/
```

```
MyUppApps/Fam_a3.3/Fam_a3.3.cpp" -o
```

```
"/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/Fam_a3.3.o"
```

```
g++-4.1 -c -I"/media/hda6/MyUppApps" -I"/media/hda6/upp/uppsrc" -I"/usr/include/freetype2"
```

```
-I"/usr/include/gtk-2.0"
```

```

-I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include" -I"/usr/include/cairo"
-I"/u
sr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC -DflagSHARED -DflagLINUX
-DbmYEAR=2008 -DbmMO
NTH=9 -DbmDAY=7 -DbmHOUR=23 -DbmMINUTE=47 -DbmSECOND=40 -fexceptions -Os
-finline-limit=20 -x c++ "/media/hda6/
MyUppApps/Fam_a3.3/Fam_a3.3.cpp" -o
"/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/Fam_a3.3.o"
compiled in (0:02.17)
parser.cpp
g++-4.1 -c -I"/media/hda6/MyUppApps" -I"/media/hda6/upp/uppsrc" -I"/usr/include/freetype2"
-I"/usr/include/gtk-2.0"
-I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include" -I"/usr/include/cairo"
-I"/u
sr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC -DflagSHARED -DflagLINUX
-DbmYEAR=2008 -DbmMO
NTH=9 -DbmDAY=7 -DbmHOUR=23 -DbmMINUTE=47 -DbmSECOND=40 -fexceptions -Os
-finline-limit=20 -x c++ "parser.cpp"
-o "/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/parser.o"
g++-4.1 -c -I"/media/hda6/MyUppApps" -I"/media/hda6/upp/uppsrc" -I"/usr/include/freetype2"
-I"/usr/include/gtk-2.0"
-I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include" -I"/usr/include/cairo"
-I"/u
sr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC -DflagSHARED -DflagLINUX
-DbmYEAR=2008 -DbmMO
NTH=9 -DbmDAY=7 -DbmHOUR=23 -DbmMINUTE=47 -DbmSECOND=40 -fexceptions -Os
-finline-limit=20 -x c++ "parser.cpp"
-o "/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/parser.o"
compiled in (0:04.20)
scanner.cpp
g++-4.1 -c -I"/media/hda6/MyUppApps" -I"/media/hda6/upp/uppsrc" -I"/usr/include/freetype2"
-I"/usr/include/gtk-2.0"
-I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include" -I"/usr/include/cairo"
-I"/u
sr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC -DflagSHARED -DflagLINUX
-DbmYEAR=2008 -DbmMO
NTH=9 -DbmDAY=7 -DbmHOUR=23 -DbmMINUTE=47 -DbmSECOND=40 -fexceptions -Os
-finline-limit=20 -x c++ "scanner.cpp"
-o "/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/scanner.o"
g++-4.1 -c -I"/media/hda6/MyUppApps" -I"/media/hda6/upp/uppsrc" -I"/usr/include/freetype2"
-I"/usr/include/gtk-2.0"
-I"/usr/include/glib-2.0" -I"/usr/lib/glib-2.0/include" -I"/usr/lib/gtk-2.0/include" -I"/usr/include/cairo"
-I"/u
sr/include/pango-1.0" -I"/usr/include/atk-1.0" -DflagMAIN -DflagGCC -DflagSHARED -DflagLINUX
-DbmYEAR=2008 -DbmMO
NTH=9 -DbmDAY=7 -DbmHOUR=23 -DbmMINUTE=47 -DbmSECOND=40 -fexceptions -Os
-finline-limit=20 -x c++ "scanner.cpp"
-o "/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/scanner.o"

```

compiled in (0:02.64)

Fam_a3.3: 5 file(s) built in (0:08.83), 1766 msecs / file, duration = 9766 msecs

Linking...

```
g++-4.1 -o "/media/hda6/MyUppApps/out/GCC.Shared/Fam_a3" -Wl,-s -L"/usr/X11R6/lib"
-Wl,-O,2 "/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/Fam_a3.3.o"
"/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/parser.o" "/media/hda
6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/scanner.o" -Wl,--start-group -lpthread -ldl -lz
"/media/hda6/MyUppApps/out/Core/GCC.Shared/Core.a" "/media/hda6/MyUppApps/out/plugin/pcre/GCC.Shared/pcre.a"
-Wl,--end-group
g++-4.1 -o "/media/hda6/MyUppApps/out/GCC.Shared/Fam_a3" -Wl,-s -L"/usr/X11R6/lib"
-Wl,-O,2 "/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/Fam_a3.3.o"
"/media/hda6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/parser.o" "/media/hda
6/MyUppApps/out/Fam_a3.3/GCC.Main.Shared/scanner.o" -Wl,--start-group -lpthread -ldl -lz
"/media/hda6/MyUppApps/out/Core/GCC.Shared/Core.a" "/media/hda6/MyUppApps/out/plugin/pcre/GCC.Shared/pcre.a"
-Wl,--end-group
Exitcode: 0
/media/hda6/MyUppApps/out/GCC.Shared/Fam_a3 (622112 B) linked in (0:01.05)
```

OK. (0:11.00)

"/media/hda6/MyUppApps/out/GCC.Shared/Fam_a3"

/media/hda6/MyUppApps/out/GCC.Shared/Fam_a3

Error executing /media/hda6/MyUppApps/out/GCC.Shared/Fam_a3

Exitcode: 1006

I'm really glad to see, that it's not a bug in thede I can just keep it in the right order (which is BTW any where parser.y is above Fam_a3.3.cpp), it's no problem. Sorry for taking your time and thanks for pointing me in the right way

Honza

Subject: Re: Different results from same code&settings

Posted by [mr_ped](#) on Mon, 08 Sep 2008 06:27:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Glad I did help, although I have no idea how, as I didn't mention order of compilation.
Thanks for sharing the cause, I'm again a tad more experienced too.

I still wonder what was the cause of the crash in the code? Did the Fam*.cpp include some old things which were supposed to be updated by parser.y ahead of it's compilation, which broke ABI and caused it to crash? Or what was it? Interesting behavior.

Subject: Re: Different results from same code&settings

Posted by [mirek](#) on Mon, 08 Sep 2008 06:39:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Do you have any global constructors?

Mirek

Subject: Re: Different results from same code&settings

Posted by [dolik.rce](#) on Mon, 08 Sep 2008 12:35:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

mr_ped wrote on Mon, 08 September 2008 08:27Glad I did help, although I have no idea how, as I didn't mention order of compilation.

You made me check if the files were corrupted Then it was easy to locate the only difference is in ordering of files

luzr wrote on Mon, 08 September 2008 08:39Do you have any global constructors?

Nothing I'm aware of. But it's possible that there's something in the jungle of code generated by Bison. I'm not really skilled programmer and I admit that I don't understand most of the generated code (This is in fact my first work with Bison&Flex - and I hope last too) Anyway, I can see how a global constructor could cause this behaviour. In fact it crashes somewhere in the middle of Flex generated scanner.cpp when creating an input buffer, it's possible that there is something uninitialized.

Thanks once more for your advices.

Honza
