Subject: Unability to debug
Posted by Wlad on Tue, 20 Aug 2013 10:26:36 GMT
View Forum Message <> Reply to Message

I did not use U++ aprox 1.5 years.
Now I tried to install several builds: stable version (5485) and last nightly builds...

My system was not changed. XP SP3.
But earlier they worked in debugging project very nice.
Now I obtain GPF just after starting in Debug Mode or IDE is trying to show something like grey pannel beneath to editor panel, shows register panel to the right of editor panel. ANd that's all.
Process Dispetcher shows that gdb is working and wait something.

I have tried to setup several MinGWs (TDM-GCC-4.7.1-2, from Qt sdk, WxPack and gcc from wxWidgets installation) - the result remains the same.

Do anibody have the same sad picture?
And if yes then how to get well-working IDE?

Subject: Re: Unability to debug
Posted by dolik.rce on Tue, 20 Aug 2013 10:55:26 GMT
View Forum Message <> Reply to Message

Welcome back Wlad

I'm not sure how it is on windows, but theide gained new gdb frontend some time ago. Can you check what is set as "GDB Debugger interface" in Setup > Environment > IDE? Try experimenting with both values, I believe the legacy should work in any case...

Best regards,
Honza

Subject: Re: Unability to debug
Posted by Wlad on Tue, 20 Aug 2013 13:09:00 GMT
View Forum Message <> Reply to Message

Quote:Welcome back Wlad
TNX you!

About "GDB Debugger interface":

I tried both of them with different minGWs.

"legacy" choice is "a little bit better"... It means I do not obtain GPF just after the F5 pressing (and neccesary preparing under the hood)

"new" gives "iron(y)" GPF at 100%!

I thought I would not meet Windows (as developer) again any more!!!!
But my new project/work requires it and I have already ripped out all the paoyt!!!!!!

Subject: Re: Unability to debug
Posted by dolik.rce on Tue, 20 Aug 2013 13:31:53 GMT
View Forum Message <> Reply to Message

Hm, bad to hear that

What about using MSVC? From what I heard, it has better debugger anyways... (I never used it myself)

Honza

Subject: Re: Unability to debug
Posted by busiek on Tue, 20 Aug 2013 15:20:15 GMT
View Forum Message <> Reply to Message

Because thread about debugger was raised, I have to confirm that there are some problems.

For me debugging using new interface is not usable. For GUI application showing backtrace (from drop list) lasts few minutes which makes it unusable to walk up or down through backtrace. I have to switch to legacy, but it is not convenient. It also seems to work fast enough for console applications. My environment is Ubuntu 13.04 64bit and the newest U++ from svn.

Subject: Re: Unability to debug
Posted by unodgs on Tue, 20 Aug 2013 16:50:32 GMT
View Forum Message <> Reply to Message

busiek wrote on Tue, 20 August 2013 11:20Because thread about debugger was raised, I have to confirm that there are some problems.

For me debugging using new interface is not usable. For GUI application showing backtrace (from drop list) lasts few minutes which makes it unusable to walk up or down through backtrace. I have to switch to legacy, but it is not convenient. It also seems to work fast enough for console applications. My environment is Ubuntu 13.04 64bit and the newest U++ from svn.

I can only confirm that. New debugger is very nice but very slow and crashes in mt apps from time to time. The best option so far is a legacy mode with MSVC.

Subject: Re: Unability to debug
Posted by nlneilson on Tue, 20 Aug 2013 19:38:19 GMT
View Forum Message <> Reply to Message

With MSVC on Win from my experience it depends on the version of MSVC and probably how it is setup.

As long as I can debug with MSVC 8, 9, 10 12 or 13 I have not worried about it. Usually with a bit of time tinkering it will debug OK with any version but don't see a reason for doing that.

It is usually my error in some way.

Subject: Re: Unability to debug
Posted by Wlad on Wed, 21 Aug 2013 06:44:22 GMT
View Forum Message <> Reply to Message

I do not use MSVC at all.
I like GCC's features and syntax. Plus I use GCC because of cross development. Of course you will agree it is better to work in the same environment and implementation of a language starting from low level microcontroller code and up to PC GUI application!

Subject: Re: Unability to debug
Posted by Shire on Sun, 29 Sep 2013 13:59:59 GMT
View Forum Message <> Reply to Message

I confirm that.
New debugger interface, GDB_MI2, fully unusable and buggy. On Linux and Windows.

Subject: Re: Unability to debug
Posted by mirek on Thu, 26 Dec 2013 10:47:24 GMT
View Forum Message <> Reply to Message

I have just fixed significant bug in MI2 debugger, perhaps it will behave better now (also, it now better supports console).

Subject: Re: Unability to debug
Posted by Shire on Fri, 31 Jan 2014 13:44:26 GMT
View Forum Message <> Reply to Message

GDB_MI2 crashes when debugging by TheIDE built with MSC compiler.
(Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 16.00.40219.01 for 80x86, for me).

Debugger crashes at this harmless place:
uppsrc/ide/Builders/Gdb_MI2.cpp

```
MIValue Gdb_MI2::MICmd(const char *cmdLine)
{ // ...
 return ReadGdb();
}
```

Return by value operation destructs MIValue instance. Let's see at MIValue class:
uppsrc/ide/Builders/MIValue.h

```
MIValue &operator=(pick_ MIValue &v);
MIValue &operator=(String const &s);

MIValue(MIValue pick_ &v);
MIValue(String const &s);

operator String&() { return Get(); }
operator const String &() const { return Get(); }
```

After some debugging I found fundamental difference
uppsrc/Core/Defs.h

```
#ifdef COMPILER_MSC
#define pick_
#else
#define pick_ const
#endif
```

Smart Microsoft compiler, when it can not find copy constructor of MIValue (because _pick is not const), copies instance via const String& conversion:

```
return MIValue((const String&)ReadGdb());
```

May be, this method was preferred by compiler because constantness of returning operator.
Solution is make String& constuctor explicit and replace operator= with Set(String&) method.
Patch is attached.

Such compiler behavior causes floating bugs. I mean this cause must be at "Pick behavior explained" article.

File Attachments

1) gdb_fix.diff, downloaded 496 times