Subject: GDB_MI2 debugger fixes
Posted by mdelfede on Fri, 07 Feb 2014 13:59:11 GMT
View Forum Message <> Reply to Message

I made many improvements on debugger; now it should run smoothly and consume much fewer cpu.

Please test it deeply and tell me if you've got any problems.

Ciao

Massimo

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Fri, 07 Feb 2014 14:38:47 GMT
View Forum Message <> Reply to Message

Hello Massimo,

It seems that now IDE has got big lag when it is in GDB_MI2 mode.

How to reproduce?
1. Start debugging in GDB_MI2 mode.
2. Click on any ide element. (It takes a lot of time to activate any of the ide elements such as CodeEditor etc.. It seems that lag time is around one second). So, editing some code in debug mode is difficult.

P.S.
Now, Locals & Watches tabs don't lock ide.

Sincerely,
klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Fri, 07 Feb 2014 15:37:49 GMT
View Forum Message <> Reply to Message

Yep, I see.... that's because of tooltip handling.
I'll try to find a better way and, if none, I'll disable it.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Fri, 07 Feb 2014 15:55:45 GMT
View Forum Message <> Reply to Message

Hi klugier,

could you please test it now ? Still not perfect, but should be faster.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Fri, 07 Feb 2014 16:16:48 GMT
View Forum Message <> Reply to Message

Hello Massimo,

It seems that it is slow as it was before. Moreover, the lag prevents me to select text in CodeEditor.

Sincerely,
klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Fri, 07 Feb 2014 16:41:05 GMT
View Forum Message <> Reply to Message

Ok, removed tooltips by now.... could you please re-test if it's ok now ?

BTW, an initial lag (after stepping/running) is foreseen, it is updating the variable display when idle.
I'll try anyways to shorten that one too.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Fri, 07 Feb 2014 17:11:57 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Now, ide works at normal speed, but as you wrote - initial lag can be observed.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sat, 08 Feb 2014 07:36:56 GMT
View Forum Message <> Reply to Message

Now all should be fixed.... no lags and tooltips working.

BTW, tooltips show ONLY values for local variables, having them evaluate arbitrary expressions is too time expensive.

Also 'autos' pane shows just local variables near cursor, not global or member ones.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sat, 08 Feb 2014 13:18:44 GMT
View Forum Message <> Reply to Message

Hello Massimo,

I would like to noticed that debbuger panel loading time is relatively large (Application works, but panel isn't loaded).

Moreover moving between tabs: "Autos", "Locals", "Watches" & "Explorer" also takes "a long" time (lag effect).

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sat, 08 Feb 2014 13:24:05 GMT
View Forum Message <> Reply to Message

Panels are loaded after 500 ms after debugger goes idle, to allow to step with fast speed without

unneeded delay.
Anyways, if you step fast you don't have time to read panels and, after you stop stepping I guess that half a second delay for update is more than acceptable.....

Switching panes should have no lag at all.... which kind of lag do you have ?

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sat, 08 Feb 2014 14:16:42 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Quote:
Panels are loaded after 500 ms after debugger goes idle, to allow to step with fast speed without unneeded delay.
Anyways, if you step fast you don't have time to read panels and, after you stop stepping I guess that half a second delay for update is more than acceptable.....

Before your updates it loaded super fast. Moreover on Windows with pdb, panels are loding a lot of faster.

Quote:
Switching panes should have no lag at all.... which kind of lag do you have ?

On my computer switching between panes takes exactly 3(+-1) seconds even if debugging is finished.

GDB & GCC information:
GNU gdb (GDB) 7.6.1-ubuntu
g++-4.8.real (Ubuntu/Linaro 4.8.1-10ubuntu9) 4.8.1

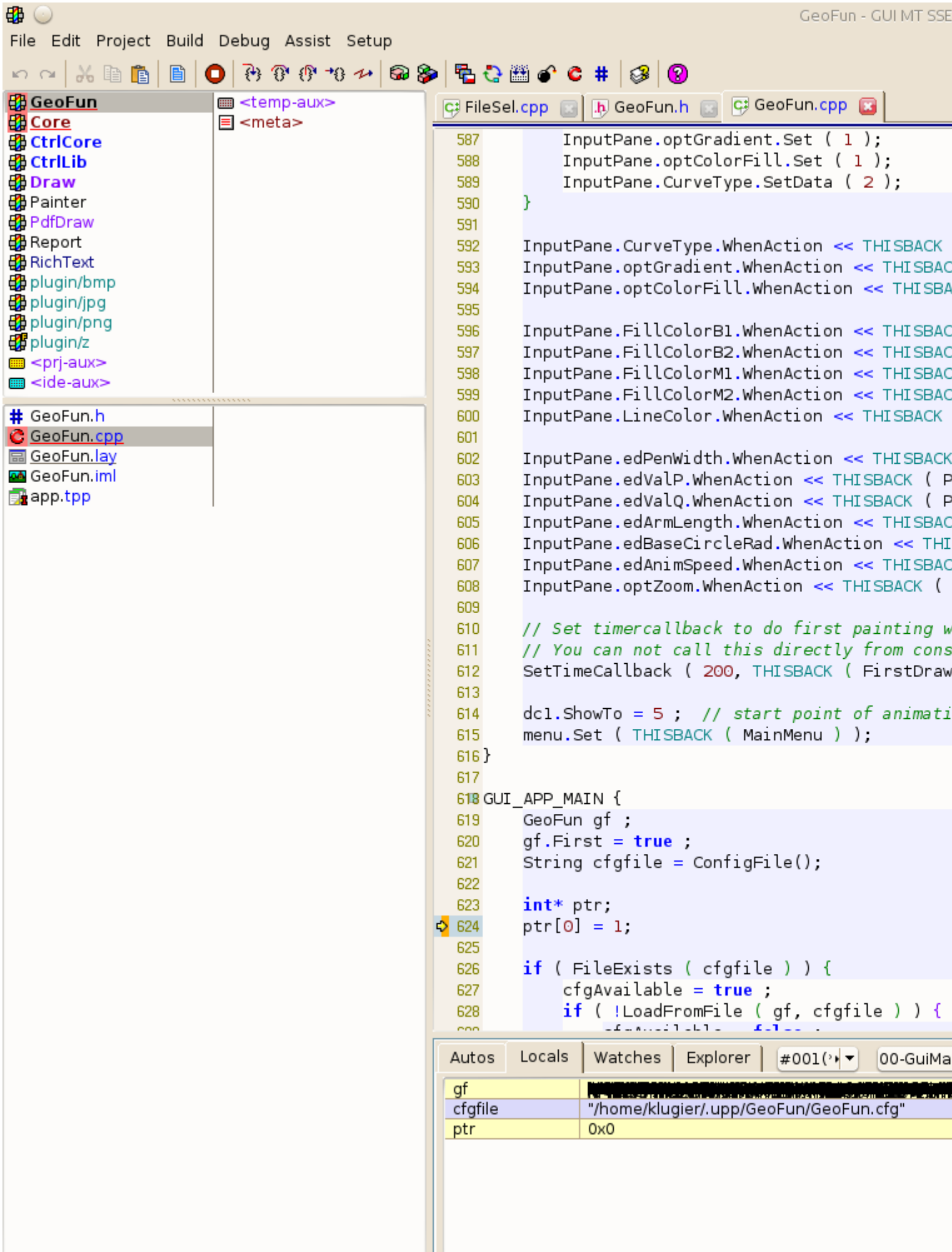
P.S.
I have found nice artifact in "Locals" pane.

P.S. 2
IDE is compiled with MT mode.

Sincerely,
Klugier

File Attachments

1) DebuggingArtifact.png, downloaded 605 times


GeoFun - GUI MT SSE
File Edit Project Build Debug Assist Setup
GeoFun
Core
CtrlCore
CtrlLib
Draw
Painter
PdfDraw
Report
RichText
plugin/bmp
plugin/jpg
plugin/png
plugin/z
<prj-aux>
<ide-aux>
<temp-aux>
<meta>
GeoFun.h
GeoFun.cpp
GeoFun.lay
GeoFun.iml
app.tpp
FileSel.cpp
GeoFun.h
GeoFun.cpp
587 InputPane.optGradient.Set (1);
588 InputPane.optColorFill.Set (1);
589 InputPane.CurveType.SetData (2);
590 }
591
592 InputPane.CurveType.WhenAction << THISBACK
593 InputPane.optGradient.WhenAction << THISBAC
594 InputPane.optColorFill.WhenAction << THISBA
595
596 InputPane.FillColorB1.WhenAction << THISBAC
597 InputPane.FillColorB2.WhenAction << THISBAC
598 InputPane.FillColorM1.WhenAction << THISBAC
599 InputPane.FillColorM2.WhenAction << THISBAC
600 InputPane.LineColor.WhenAction << THISBACK
601
602 InputPane.edPenWidth.WhenAction << THISBACK
603 InputPane.edValP.WhenAction << THISBACK (P
604 InputPane.edValQ.WhenAction << THISBACK (P
605 InputPane.edArmLength.WhenAction << THISBAC
606 InputPane.edBaseCircleRad.WhenAction << THI
607 InputPane.edAnimSpeed.WhenAction << THISBAC
608 InputPane.optZoom.WhenAction << THISBACK (
609
610 // Set timercallback to do first painting w
611 // You can not call this directly from cons
612 SetTimeCallback (200, THISBACK (FirstDraw
613
614 dc1.ShowTo = 5 ; // start point of animati
615 menu.Set (THISBACK (MainMenu));
616 }
617
618 GUI_APP_MAIN {
619 GeoFun gf ;
620 gf.First = true ;
621 String cfgfile = ConfigFile();
622
623 int* ptr;
624 ptr[0] = 1;
625
626 if (FileExists (cfgfile)) {
627 cfgAvailable = true ;
628 if (!LoadFromFile (gf, cfgfile)) {
Autos
Locals
Watches
Explorer
#001(
00-GuiMa
gf
cfgfile
"/home/klugier/.upp/GeoFun/GeoFun.cfg"
ptr
0x0

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sat, 08 Feb 2014 15:48:04 GMT
View Forum Message <> Reply to Message

klugier wrote on Sat, 08 February 2014 15:16Hello Massimo,

Before your updates it loaded super fast. Moreover on Windows with pdb, panels are loding a lot of faster.

I don't notice any delay in loading, besides the foreseen 500 ms delay on loading data in explore panes (locals, autos, etc..),
which is needed to have fast stepping.

Quote:
On my computer switching between panes takes exactly 3(+-1) seconds even if debugging is finished.

On my computer the delay is zero, when switching panes.... are you sure you've got last svn update ?
And I've no artifact at all.
I'm on Ubuntu 13.10, gcc 4.8.1 too, and I guess same GDB as yours.
The only difference is that my IDE is compiled WITHOUT Mt.

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sat, 08 Feb 2014 20:04:56 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Quote:
On my computer the delay is zero, when switching panes.... are you sure you've got last svn update ?

I compiled upp on without gadgets like "X11 MT SSE2". I also removed all previous compilation objects files. Still not luck. Delay is felt.

I updated ide source files with following command:

svn checkout http://upp-mirror.googlecode.com/svn/trunk/uppsrc/ide

If you want to make additional check I enclose my GDB2_MI2.cpp file.

Edit:

Please see 515 line(GDB_MI2.cpp):
Quote:
Sleep(20);

After commenting this line switching between panes back to normal state, but ide crashes.
Personally, I think that something must be wrong in Gdb_MI2::ReadGdb(bool wait) method.

Sincerely,
Klugier

File Attachments

1) Gdb_MI2.cpp, downloaded 408 times

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sat, 08 Feb 2014 20:47:26 GMT
View Forum Message <> Reply to Message

Hello Massimo,

It seems that changing retries initial value from "ReadGdb" method to previous value did the trick.
(Gdb_MI2.cpp - line 508):

int retries = 4;

insted of

int retries = 3 * 50;

It seems that I have got the latest Ultimate++ version. MT mode does not cause this issue also.
Max 3 seconds (3 * 50 * 20) waiting time is explicitly written to the code.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sun, 09 Feb 2014 00:37:18 GMT
View Forum Message <> Reply to Message

Hi,

ReadGdb blocks, in blocking path, for exactly 3 seconds, so

3*50*20 ms

Which is right, imho, for the 'almosto blocking' path.

This has nothing to do with panes delay.
It's just a way to wait for gdb to answer to a command... normally it responds in less than 20 ms, 3 seconds is a maximum as when gdb returns python exceptions usually it needs almost 1 second to recover.

Again, this has nothing to do with delay in panes switching, which is totally unrelated with gdb communication.

I think you've got some problem with your gdb... but I can't replicate it.

Just to explain you the path:

1) When I send a command to gdb, I wait for response, which is completed when I receive a '(gdb)' string at end.
To be sure to get the response, I put a limit of 3 seconds on it, but as I said, normally it answers in 20 ms or less.

2) Switching panes is totally unrelated, as panes works on cached data, so do not need to issue additional gdb commands.

3) Panes updating happens 500 ms AFTER last gdb command.
So, if you step quickly, panes do not update before you stop stepping.

If you like, you can send me your compiled ide, so I can test it.
Send to this address : maxx5@veneto.com

ciao

Massimo

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 09 Feb 2014 12:32:47 GMT
View Forum Message <> Reply to Message

Hello Massimo,

I have sent you my ide.

I compiled pure version from Ultimate++ website. It seems that looding time is now OK, but switching between panes while debugging is still slow. When you finished debugging it works at normal speed.

But there is one more big issue. IDE does not automatically close debugg widgets when debugging process is over. In previous builds it worked properly.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sun, 09 Feb 2014 13:40:13 GMT
View Forum Message <> Reply to Message

Hi,

I've just tested your build, but here it works perfectly.
No lags switching panes (I don't know what do you mean with "while debugging", panes are not there when debugger is not running....) and on application exit the debugger terminates correctly.

Are you SURE that you're running THIS build and not another, installed one on your system ???

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 09 Feb 2014 15:20:28 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Quote:
Are you SURE that you're running THIS build and not another, installed one on your system ???

It seems that gtk backend on KDE cause this issue. When I switch to X11 debug mode closes correctly.

So this is definitly not Gdb_MI2 issue.

Quote:
I've just tested your build, but here it works perfectly.
No lags switching panes (I don't know what do you mean with "while debugging", panes are not

there when debugger is not running....) and on application exit the debugger terminates correctly.

How to reproduce?
1. Start debugging.
2. Back to IDE.
3. Try to switch between following panes: Autos, Locals, Watches and Explorer. (Debug mode)
4. Now, you can stop debugging.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sun, 09 Feb 2014 15:23:10 GMT
View Forum Message <> Reply to Message

Ah, I'm not running KDE, I'm on gnome and I've no KDE installed here.... so I can't reproduce.
Anyways, I'm on GTK backend too.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 09 Feb 2014 15:26:10 GMT
View Forum Message <> Reply to Message

Quote:
Ah, I'm not running KDE, I'm on gnome and I've no KDE installed here.... so I can't reproduce.
Anyways, I'm on GTK backend too.

But, I am almost sure that lag between switching panes in debug mode is not KDE related.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by Oblivion on Sun, 09 Feb 2014 17:25:02 GMT
View Forum Message <> Reply to Message

klugier wrote on Sun, 09 February 2014 17:20Hello Massimo,

Quote:
Are you SURE that you're running THIS build and not another, installed one on your system ???

It seems that gtk backend on KDE cause this issue. When I switch to X11 debug mode closes correctly.

So this is definitly not Gdb_MI2 issue.

Quote:
I've just tested your build, but here it works perfectly.
No lags switching panes (I don't know what do you mean with "while debugging", panes are not there when debugger is not running....) and on application exit the debugger terminates correctly.

How to reproduce?
1. Start debugging.
2. Back to IDE.
3. Try to switch between following panes: Autos, Locals, Watches and Explorer. (Debug mode)
4. Now, you can stop debugging.

Sincerely,
Klugier

I can confirm this. I am also working on KDE 4.12 (latest Upp SVN 6888). When working with TheIDE on GTK+ backend, sometimes debug mode (GDB_MI2) does not quit. Panes stays where they are.
Also,
1) Is the disasm removed from the new GDB_MI2 interface, because the right pane where a disassembler should be, is almost empty (Shows only stack pointer, etc.) ? It only works with the legacy gdb interface.
2) There is no yellow pointer arrow (which shows the current position) on breakpoints with the new interface. Is it removed too? If so, it makes debugging a lot harder, since I have to rely on intuition.

Other than these, it seems really improved, thanks for your work!
Regards.

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sun, 09 Feb 2014 17:30:33 GMT
View Forum Message <> Reply to Message

Hi Oblivion,

I'm still working on some other gdb improvements... but I've got none of the problem you're having.

In gnome, gtk backend, all is working perfectly, the disasm pane too.

Just wait that I'm finshed with current development, then I'll try to look at your problems.
In the meanwhile, if you can find the cause you're wellcome.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 09 Feb 2014 18:36:25 GMT
View Forum Message <> Reply to Message

Hi Massimo,

On Ubuntu 13.10 GDB_MI2 works perfect (I have tested with VirtualBox).

But on all my KDE machines debugging has got "lag panes" issue.

Massimo can you test IDE on Kubuntu or other KDE distribution via for example VirtualBox?

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by Oblivion on Mon, 10 Feb 2014 00:38:45 GMT
View Forum Message <> Reply to Message

Hello Massimo,

I dug through the TheIDE's code and found the main reason behind the two problems I've mentioned above.
Both problems seem to be a result of a localization/internationalization issue.

You see, My upp dir is under a directory which has some Turkish letters in its name. Both Gdb_MI2::SyncIde() and Gdb_MI2::SyncDisas() methods call the MICmd() method to fill MIValue structures, which seem to have a "file" (tuple?) member in it. But calling the MICmd() method returns a file path and if that path has non-ascii characters, it converts them to a number value (utf or unicode?).

Please find the attached screenshot, which I believe can explain the problem better. In the screenshot, as you will see, because the path doesn't actually exist (the path should not be ".../Geli305237tirme/Ultimate++/...", theide does not call disas.Add().

If I move the Ultimate++ directory under, say, /home/user/ directory everything works fine.

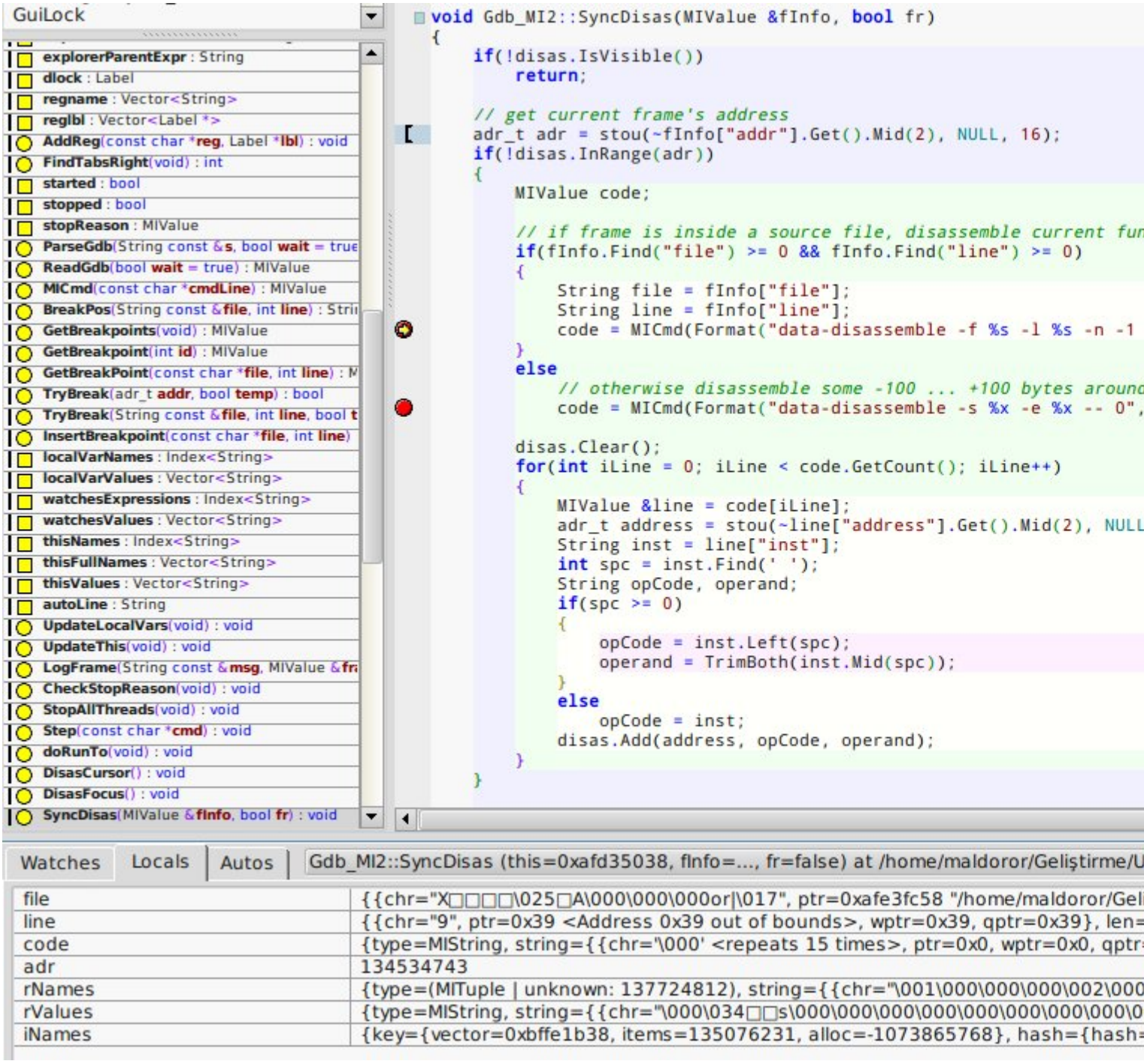
Shouldn't these "localized path" issues be taken care of by theide? Or is this a GDB issue?

(After more investigation: It seems that this is a gdb issue. Gdb uses the numeric representations of non-ascii characters in paths.)

Regards.

File Attachments

1) screenshot.jpg, downloaded 670 times



Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Mon, 10 Feb 2014 08:13:30 GMT
View Forum Message <> Reply to Message

Hi Oblivion,

this is quite weird.... but I stumbled on a problem with gdb and localization too. It returns (normally, without a prettyprinter python code for it...) decimal number localized, so with ',' instead

of '.' in italian, which made parsing of expression data almost impossible.
As I don't use special characters in my filenames I've never seen your problem.
I'll test it when finished with my 'this' explorer pane, probably in a couple of days.

p.s.: the MIValue is a kind of value class that is able to parse and hold gdb data responses. 'tuple' is the equivalent (almost...) of python tuples : { a b }, nothing to do with localization. Probably the problem is inside path handling.

Klugier, about your 'lag' problems, I'll see if I find some time to setup a virtualbox with kde. I don't want to clobber my ubuntu setup with another shell
Can't you try to debug it to locate the problem ?

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Oblivion on Mon, 10 Feb 2014 15:21:32 GMT
View Forum Message <> Reply to Message

Hello Massimo,

I've done some search on the net, and come by this thread. Same problem and a small discussion, you may want to read it:

http://sourceware.org/ml/gdb/2010-08/msg00119.html

These guys seem to hit the same problem. It seems that the GDB MI interface converts the UTF-8 characters to 7 bit + octal representation. So either we have to take care of it, and manually parse the path to UTF-8 or there is a command which can do it for us that I don't know.

Regards.

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Tue, 11 Feb 2014 20:58:51 GMT
View Forum Message <> Reply to Message

Hi all,

I've just uploaded a major debugger revision.

Changes:

1) The lag switching tabs should be gone

2) Stepping speed greatly enhanced
3) Added explorer pane for 'this' variable
4) Removed (disabled by now..) python prettyprinters, they will be replaced by c++ modules
5) Removed usage of GDB variables... they're both too slow and too buggy... now variabled are evaluated by gdb command '-data-evaluate-expression', by far faster even if much more difficult to parse results

By now, of Upp types, just the String is pretty printed, all other types are accessible by internal variables.
I'll try to extend it for most common types, probably implementing some plugin mechanics.

Please test and give suggestions!

Ciao

Max

p.s.: still experimental, even if better than before, so expect some bugs.....

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Thu, 13 Feb 2014 00:30:02 GMT
View Forum Message <> Reply to Message

Some more enhancements :

Now type evaluators for most common Upp types : String, Vector, VectorMap, Array and ArrayMap.
You can see it at work inside the 'this' pane.

Please test and report any problem.... the code is still quite experimental.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Oblivion on Thu, 13 Feb 2014 10:16:27 GMT
View Forum Message <> Reply to Message

Quote:
Changes:

1) The lag switching tabs should be gone
2) Stepping speed greatly enhanced

3) Added explorer pane for 'this' variable
4) Removed (disabled by now..) python prettyprinters, they will be replaced by c++ modules
5) Removed usage of GDB variables... they're both too slow and too buggy... now variabled are evaluated by gdb command '-data-evaluate-expression', by far faster even if much more difficult to parse results

Hi Massimo,
Thanks for your work!

Currently, with TheIDE (6913), on Arch Linux (kernel v: 3.12.10, KDE 4.12, i686, with no SSE2):

1) Yes, lags are gone. At least in X11 mode. GTK mode in theIDE is so laggy in general that I can't use, so I can't say if there's any noticable speed up there (I mean, TheIDE (GTK), especially the code editor and debugger is laggy to an unusable degree on my machine, but it has nothing to do with debugger per se).

2) Yes, stepping speed is greatly enhanced. Only the first initialization takes a relatively long time (4-5 seconds).

3) This pane and type evaluators seem to work, and is handy.

I've encountered some seemingly random dead locks (happens when I run/exit the same apps several times in debug mode). Unfortunately I couldn't determine the cause yet.

Regards.

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Thu, 13 Feb 2014 13:14:06 GMT
View Forum Message <> Reply to Message

Hello Massimo and Oblivion,

Quote:
2) Yes, stepping speed is greatly enhanced. Only the first initialization takes a relatively long time (4-5 seconds).

Gdb initialization on KDE still takes a lot of time. On Ubuntu it is super fast.

Moreover, now only the first panes switch is laggy. After that everything seems to works OK on KDE.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Thu, 13 Feb 2014 13:23:51 GMT
View Forum Message <> Reply to Message

Hi Oblivion and Krugier,

I just noticed slow down when there are many arrays/maps to in 'this' inspector.... gdb needs to be called in several times and that's slow.
I've to decide if it's better to not show first container's elements or to just abort their operation if some event is in queue.
The first is easy but I find handy to see first elements in arrays and maps without having to inspect the variable... the second seems quite hard to achieve.
The alternative would be to add a checkbox on settings page for it.... what is your opinion ?
In my PC for a really complex 'this' class it takes less than half a second without containers evaluation, and more than 5-6 seconds if I have some 10 containers to evaluate, which is clearly too much time.

Ah, tooltips are disabled by now, I'm still implementing them in a fast way, it'll be ready in short time.

About general lags, I have theide in GTK mode and I don't notice any of them....

Ciao

Max

p.s.: Klugier, the 'first pane switching' is not lagging, probably you've got many containers in your class. It's the evaluation that is slow. You'll notice the lag also when stepping, after each quick step (about 0.5 seconds after last step, when re-evaluation begins).

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Wed, 19 Feb 2014 10:41:13 GMT
View Forum Message <> Reply to Message

Hi,

I did a big refactoring on debugger code (linux side).
It's still single-threaded (MT is on the way...) but most lags should be gone.

Please test!

I'll do some more refactoring on explore pane and quick watch, then try to go multithreading in order to remove remaining small lags.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 23 Feb 2014 00:02:47 GMT
View Forum Message <> Reply to Message

Hello Massimo,

It seems, that GDB initializing is slower than previous. (Kubuntu 13.10 - KDE 4.11).

I measured initializing time using my watch. It takes around 15 seconds (5 times for 3 seconds!), so it seems that it consumes too much time. And I don't know how to fix it.

Like I noticed several post ago. This code generates slow KDE initializing time.

```
MIValue Gdb_MI2::ReadGdb(bool wait)
{
 String output, s;
 MIValue res;

 if(wait)
 {
  // blocking path
  // waits for 3 seconds max, then return empty value
  // some commands (in particular if they return python exceptions)
  // have a delay between returned exception text and command result
  // so we shall wait up to the final (gdb)
  int retries = 3; // 13 * 50; // <- This variable is too big for KDE. Max retries is always executed at startup.
  while(dbg && retries--)
  {
   dbg->Read(s);
   output += s;
   if(TrimRight(s).EndsWith("(gdb)"))
    break;
   Sleep(20);
   continue;
  }
 }
 else if(dbg)
  dbg->Read(output);
 if(output.IsEmpty())
  return res;
 return ParseGdb(output);
}
```

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sun, 23 Feb 2014 08:29:17 GMT
View Forum Message <> Reply to Message

Hi Kligier,

this path is needed to wait for correct gdb response.
I don't know why it's so slow on kde, I don't have it right here.

Anyways, I'm refactoring most of MI2 debugger and adding multithreading support, so it should not slow the interface.

If you want to help about your problem, just add this line after dbg->Read(s); one :

RLOG(s);

Run the debugger and post the IDE log contents.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 23 Feb 2014 13:44:29 GMT
View Forum Message <> Reply to Message

Hello Massimo,

I enclose log file.

Sincerely,
Klugier

File Attachments
1) theide.log, downloaded 391 times

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Tue, 25 Feb 2014 16:51:48 GMT
View Forum Message <> Reply to Message

Hi,

I refactored all the stuff again, and not there's an option to use multithreading.

So, please test it now, building with an option containing MT.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sat, 01 Mar 2014 17:47:39 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Now, Debugger starts super fast on KDE. I tested with "GTK MT IDE".

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Sat, 01 Mar 2014 20:55:09 GMT
View Forum Message <> Reply to Message

Ok thanks.... I think now it's almost perfect.

I have still to check a couple of things and it's finished.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Oblivion on Thu, 06 Mar 2014 23:52:58 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Thank you very much for your efforts.
I too can confirm that the latest TheIDE (MT) on KDE (4.12.3) is now very fast and very responsive (both panes and overall debugging).
The issue with escaped path names that I mentioned before is also fixed.
Keep up the good work!

Regards.

Subject: Re: GDB_MI2 debugger fixes
Posted by mdelfede on Fri, 07 Mar 2014 08:26:43 GMT
View Forum Message <> Reply to Message

Thank to you to test

It's still slow updating pane's data, but that one is a GDB limit and I can't do anything about it.... the only solution would be to take LLDB codebase and embed it directly, but that one would be an huge work.

Ciao

Max

Subject: Re: GDB_MI2 debugger fixes
Posted by Klugier on Sun, 30 Mar 2014 14:35:47 GMT
View Forum Message <> Reply to Message

Hello Massimo,

Now, I have got new problem (Latest Upp - SVN 7112). GDB_MI2 dosen't hide the interface when application ends normally. I don't know what causing this issue, but before update it works greate.

P.S.
After cleaning upp cahce, GDB starts working again.

Sincerely,
Klugier

Subject: Re: GDB_MI2 debugger fixes
Posted by busiek on Fri, 18 Apr 2014 10:52:28 GMT
View Forum Message <> Reply to Message

Hello everyone,

After recent changes in Core new debugger code doesn't compile in C++11 mode.

Subject: Re: GDB_MI2 debugger fixes
Posted by mirek on Fri, 18 Apr 2014 13:00:09 GMT
View Forum Message <> Reply to Message

busiek wrote on Fri, 18 April 2014 10:52Hello everyone,

After recent changes in Core new debugger code doesn't compile in C++11 mode.

Well, it did compile as multithreded, which is the new default. However, single threaded IDE compilation was broken. Now it is fixed.

Mirek
