

# Is there a C++ gdb GUI for Linux?

## [closed]

Asked 16 years, 3 months ago   Modified 1 month ago

Viewed 135k times



225



**Closed.** This question is seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. It does not meet [Stack Overflow guidelines](#). It is not currently accepting answers.



We don't allow questions seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. You can edit the question so it can be answered with facts and citations.

Closed 8 years ago.

[Improve this question](#)

Briefly: Does anyone know of a GUI for gdb that brings it on par or close to the feature set you get in the more recent version of Visual C++?

In detail: As someone who has spent a lot of time programming in Windows, one of the larger stumbling blocks I've found whenever I have to code C++ in Linux is that debugging anything using commandline gdb takes me several times longer than it does in Visual Studio, and

it does not seem to be getting better with practice. Some things are just easier or faster to express graphically.

Specifically, I'm looking for a GUI that:

- Handles all the basics like stepping over & into code, watch variables and breakpoints
- Understands and can display the contents of complex & nested C++ data types
- Doesn't get confused by and preferably can intelligently step through templated code and data structures while displaying relevant information such as the parameter types
- Can handle threaded applications and switch between different threads to step through or view the state of
- Can handle attaching to an already-started process or reading a core dump, in addition to starting the program up in gdb

If such a program does not exist, then I'd like to hear about experiences people have had with programs that meet at least some of the bullet points. Does anyone have any recommendations?

**Edit:**

Listing out the possibilities is great, and I'll take what I can get, but it would be even more helpful if you could include in your responses:

(a) Whether or not you've actually used this GUI and if so,

what positive/negative feedback you have about it.

(b) If you know, which of the above-mentioned features are/aren't supported

Lists are easy to come by, sites like this are great because you can get an idea of people's personal experiences with applications.

c++

linux

gdb

debugging

Share

Improve this question

Follow

edited Jun 10, 2015 at 5:31



Ciro Santilli

OurBigBook.com

380k ● 117 ● 1.3k ● 1.1k

asked Sep 17, 2008 at 1:42



HappyDude

2,606 ● 3 ● 19 ● 14


---

24 +1 for "Sites like this are great because you can get an idea of people's personal experiences." Too many people nowadays just do a list dump which IMHO completely misses the point. – [kizzx2](#) Jun 11, 2010 at 13:43

---


1 On reverse engineering:  
[reverseengineering.stackexchange.com/questions/1392/...](https://reverseengineering.stackexchange.com/questions/1392/...)  
– [Ciro Santilli OurBigBook.com](#) Jul 10, 2015 at 8:31

---

See also similar question: [What is a good unix alternative to DDD \(Data Display Debugger\)?](#) – [oHo](#) Jul 7, 2017 at 23:45 

---

See this list of gdb front-ends here:  
[sourceware.org/gdb/wiki/GDB%20Front%20Ends](https://sourceware.org/gdb/wiki/GDB%20Front%20Ends), and give

the first one in the list a shot. Looks very promising and modern and is cross-platform since it's browser-based: [gdbgui.com](http://gdbgui.com). – Gabriel Staples Jun 23, 2020 at 18:59 

---

28 Answers

Sorted by:

Highest score (default)



87

Eclipse CDT will provide an experience comparable to using Visual Studio. I use Eclipse CDT on a daily basis for writing code and debugging local and remote processes.



If you're not familiar with using an Eclipse based IDE, the GUI will take a little getting used to. However, once you get to understand the GUI ideas that are unique to Eclipse (e.g. a perspective), using the tool becomes a nice experience.

The CDT tooling provides a decent C/C++ indexer that allows you to quickly find references to methods in your code base. It also provides a nice macro expansion tool and limited refactoring support.

With regards to support for debugging, CDT is able to do everything in your list with the exception of reading a core dump (it may support this, but I have never tried to use this feature). Also, my experience with debugging code using templates is limited, so I'm not sure what kind of experience CDT will provide in this regard.

For more information about debugging using Eclipse CDT, you may want to check out these guides:

- [Interfacing with the CDT debugger, Part 2: Accessing gdb with the Eclipse CDT and MI](#)
- [CDT Debug Tutorial](#)

Share Improve this answer

Follow

edited Nov 4, 2021 at 17:57



Sean Bright

121k ● 17 ● 143 ● 149

answered Sep 17, 2008 at 2:28



Jon Ball

3,100 ● 3 ● 24 ● 24

---

10 Reading core dumps is supported. (It's referred to as the Postmortem Debugger.) – [Josh Kelley](#) Feb 2, 2009 at 1:24

---

I've had better experience in Eclipse CDT debugging than in Visual Studios. I've not given MSVC too much of a chance, though. – [notlesh](#) May 14, 2013 at 4:48

---

4 I think that [QtCreator](#) should be mentioned alongside Eclipse as it is comparable. – [nonsensickle](#) Feb 9, 2017 at 10:42

---

"Interfacing with the CDT debugger" link is gone – [milahu](#) Feb 5 at 9:45

---



80



`gdb -tui` works okay if you want something GUI-ish, but still character based.

Share Improve this answer

Follow

edited Nov 9 at 5:40



EsmaeelE

2,649 ● 7 ● 26 ● 35

answered Sep 17, 2008 at 2:08



- 
- 48 You can also go into TUI (text user interface) mode while in gdb by using the '-' command. Returning to the command prompt is Ctrl-X Ctrl-A. While in TUI mode, up, down, left, and right move through the source. Use Ctrl-P, Ctrl-N, Ctrl-F, and Ctrl-B to navigate in the command line history.  
– Ben Combee Oct 25, 2008 at 21:10

---

This is exactly what I was looking for. The power of the gdb command line with some useful views that update as I move around. – Kevin Cox Aug 13, 2014 at 21:19

- 
- 8 A .gdbinit on steroids gives you a really impressive character based GUI, with color: [github.com/cyrus-and/gdb-dashboard](https://github.com/cyrus-and/gdb-dashboard)  
– cs01 Sep 26, 2016 at 0:16 ✎

---

I quite like [gdb-dashboard](https://github.com/cyrus-and/gdb-dashboard) but have to mention [voltron](https://github.com/voltron/voltron) as well. – nonsensickle Feb 9, 2017 at 10:39

- 
- 4 @Barry thanks. For anyone curious, here is a link: [github.com/cs01/gdbgui](https://github.com/cs01/gdbgui). It has data structure display similar to DDD, a client/server architecture to easily debug remote machines with no X forwarding necessary, and ability to explore complex variables, etc.. – cs01 Oct 25, 2017 at 22:03 ✎

**58**

You won't find *anything* overlaying GDB which can compete with the raw power of the Visual Studio debugger. It's just too powerful, and it's just too well integrated inside the IDE.

For a Linux alternative, try DDD if free software is your thing.

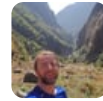


Share Improve this answer

Follow



edited Apr 20, 2016 at 15:20



Aaron McDaid

27.1k ● 10 ● 71 ● 93

answered Sep 17, 2008 at 1:45



Vicent Marti

7,295 ● 6 ● 32 ● 34

---

2 WinDBG is far superior than what Visual Studio provides. The visual studio debugger sits on top of the same architecture than WinDBG. With that said, GDB has nothing to ask when comparing it to Windbg. A graphical debugger that is better than DDD would be great. Eclipse CDT is a good alternative. I think there was a KDE UI – [bigmonachus](#) Feb 23, 2012 at 2:12

---

138 Oh look. An answer that makes a claim without backing it up at all. I also find it odd that the answer which *does not answer the question* was accepted. – [alternative](#) Mar 19, 2012 at 22:29

---

5 for anyone coming here looking for a real answer, know that ddd is garbage. The best debugger GUI for Linux I have found is IDA, which is not free software. – [bkconrad](#) Feb 7, 2013 at 18:04

---

2 Answer the question as best possible or don't say anything. Having used both, Visual Studio is more featured filled, but DDD is good enough that I don't miss VS. Sadly, DDD is showing its age. – [Alan De Smet](#) Feb 6, 2014 at 20:33

---

7 This post needs to be updated/improved. Qt Creator is awesome, on par with Visual Studio. – [vharron](#) Jul 24, 2014 at 3:13

---



Check out [Nemiver C/C++ Debugger](#). It is easy to install in Ubuntu (Developer Tools/Debugging).

51

**Update:** New link.



Share Improve this answer

edited Jan 15, 2014 at 0:41



Follow



answered Jun 1, 2011 at 5:43



[Sergey Shandar](#)

2,387 ● 18 ● 27

---

*Absolutely fabulous*, also supported with apt: 'apt-get install nemiver'. My only quibble is building from repo failed due to a 'No package gconf-2.0 found'. – [J Evans](#) Feb 5, 2015 at 13:36

---

the interface is quite clear and good to read, but it crashes sometimes while using the interface capabilities v0.9.6. It also accepts debugged application parameters, what I still cant let ddd do. – [Aquarius Power](#) Jul 11, 2018 at 16:05

---

Also available on Fedora 30 (at least). I just chanced on this question and nemiver. – [user3236841](#) Oct 18, 2019 at 18:10

---

nemiver was archived in year 2020 – [milahu](#) Feb 5 at 9:43

---





Qt Creator seems like good stuff. A colleague showed me one way set it up for debugging:

**33**



- Create a new project, "Import of Makefile-based Project".
- Point it to your root project folder (it will index sources under it, and it is impressively fast).
- Go to project settings and add a run configuration, then specify the executable you want to debug, and its arguments.
- Qt Creator seems to insist on building your project before debugging it. If you don't want that, or don't use make, just go to projects -> build (Left panel), then, on the right panel in "Build Steps", remove all the steps, including the step by default when you created the project.

That may seem like a bit much work for debugging an app I had already compiled, but it is worth it. The debugger shows threads, stacks and local variables in a similar way to Visual Studio and even uses many of the same keyboard shortcuts. It seems to handle templates well, at least `std::string` and `std::map`. Attaching to existing processes and core dumps seems to be supported, though I haven't tested it yet.

Keep in mind that I used it for less than an hour now, but I'm impressed so far.

Share Improve this answer

edited Dec 12, 2019 at 8:02

Follow



Robin Hsu

198 ● 1 ● 10

answered Aug 24, 2010 at 13:40



Soulman

3,030 ● 26 ● 21



31



I loathe the idea of Windows development, but the VC++ debugger is among the best I've seen. I haven't found a GUI front end that comes close to the VC one.

GDB is awesome once you really get used to it. Use it in anger enough and you'll become very proficient. I can whiz around a program doing all the things you listed without much effort anymore. It did take a month or so of suffering over a SSH link to a remote server before I was proficient. I'd never go back though.

DDD is really powerful but it was quite buggy. I found it froze up quite often when it got messages from GDB that it didn't grok. It's good because it has a gdb interface window so you can see what's going on and also interact with gdb directly. DDD can't be used on a remote X session in my environment (a real problem, since I'm sitting at a thin client when I do Unix dev) for some reason so it's out for me.

KDevelop followed typical KDE style and exposed EVERYTHING to the user. I also never had any luck debugging non KDevelop programs in KDevelop.

The Gnat Programming Studio (GPS) is actually quite a good front-end to GDB. It doesn't just manage Ada projects, so it's worth trying out if you are in need of a debugger.

You could use Eclipse, but it's pretty heavy weight and a lot of seasoned Unix people I've worked with (me included) don't care much for its interface, which won't just STFU and get out of your way. Eclipse also seems to take up a lot of space and run like a dog.

Share Improve this answer

answered Feb 2, 2009 at 1:29

Follow



Adam Hawes

5,449 ● 1 ● 24 ● 30

- 
- 2 Just wanted to plug my project, gdbgui ([github.com/cs01/gdbgui](https://github.com/cs01/gdbgui)). It's a client/server architecture, so works well debugging remote machines without an X session required. It also has a terminal to directly interact w/ GDB, and has visuals to display data structures, similar to DDD.  
– [cs01](#) Oct 25, 2017 at 22:01
- 

Comparisons with pros and cons based on experience. Just what this question needs. ty – [Heath Raftery](#) Apr 27, 2019 at 7:01

---



I use cgdb, simple and usefull

29

Share Improve this answer

answered Jul 15, 2009 at 11:54

Follow



andcode

390 ● 3 ● 5





16



I use DDD a lot, and it's pretty powerful once you learn to use it. One thing I would say is don't use it over X over the WAN because it seems to do a lot of unnecessary screen updates.

Also, if you're not mated to GDB and don't mind ponying up a little cash, then I would try TotalView. It has a bit of a steep learning curve (it could definitely be more intuitive), but it's the best C++ debugger I've ever used on any platform and can be extended to introspect your objects in custom ways (thus allowing you to view an STL list as an actual list of objects, and not a bunch of confusing internal data members, etc.)

Share Improve this answer

answered Sep 17, 2008 at 2:10

Follow



[Nick Bastin](#)

31.2k ● 7 ● 61 ● 78

---

I can't get DDD to work over remote X at all; just crashes out with Xlib errors on my thin client :( – [Adam Hawes](#) Feb 2, 2009 at 1:31

---

totalview looks like a viable alternative. I don't consider DDD to be better than gdb curses interface. – [deft\\_code](#) Aug 27, 2009 at 2:51

---



DDD is the GNU frontend for gdb:

<http://www.gnu.org/software/ddd/>

10

Share Improve this answer

answered Sep 17, 2008 at 1:50



Follow



[mmattax](#)

27.6k ● 42 ● 117 ● 151



I used KDbg (only works under KDE).

9

Share Improve this answer

answered Oct 25, 2008 at 19:01



Follow



[codeguru](#)

1,851 ● 2 ● 15 ● 11



---

4 KDbg is *not* limited to KDE. – [Brent Bradburn](#) May 12, 2016 at 14:55

---

That is **awesome**. I wonder where it's been all this time.

– [edmz](#) Jun 14, 2018 at 11:38



8

Check out the Eclipse CDT project. It is a plugin for Eclipse geared towards C/C++ development and includes a fairly feature rich debugging perspective (that behind the scenes uses GDB). It is available on a wide variety of platforms.



Share Improve this answer

answered Sep 17, 2008 at 1:48

Follow



user14636

398 ● 5 ● 9



Similar comfortable to the eclipse gdb frontend is the emacs frontend, tightly tied to the emacs IDE. If you already work with emacs, you will like it:

8



[GDB Emacs Frontend](#)



Share Improve this answer

edited May 23, 2017 at 12:34

Follow



Community Bot

1 ● 1



answered Jan 3, 2009 at 15:49



Johannes Schaub - litb

506k ● 131 ● 917 ● 1.2k



Qt Creator-on-Linux is certainly on par with Visual Studio-on-Windows for C++ nowadays. I'd even say better on the debugger side.

6



Share Improve this answer

answered Nov 20, 2010 at 15:47

Follow



Marv

61 ● 1 ● 1





5



I've tried a couple of different guis for gdb and have found DDD to be the better of them. And while I can't comment on other, non-gdb offerings for linux I've used a number of other debuggers on other platforms.

gdb does the majority of the things that you have in your wish list. DDD puts a nicer front on them. For example thread switching is made simpler. Setting breakpoints is as simple as you would expect.

You also get a cli window in case there is something obscure that you want to do.

The one feature of DDD that stands out above any other debugger that I've used is the data "graphing". This allows you to display and arrange structures, objects and memory as draggable boxes. Double clicking a pointer will open up the dereferenced data with visual links back to the parent.

Share Improve this answer

Follow

answered Sep 23, 2008 at 12:10



[Andrew Edgecombe](#)

40.3k ● 3 ● 38 ● 63

---

Does ddd work on programs with arguments? I could not get it to pass the arguments... – [user3236841](#) Oct 18, 2019 at 18:12

---

If I remember correctly, DDD can give you a console window directly to GDB. In the GDB console you can set the command line arguments using the "set args xyz abc" command. – [Andrew Edgecombe](#) Oct 29, 2019 at 6:05

---



5

There's one IDE that is missing in this list and which is very efficient (I've used it in many C/C++ projects without any issues): [Netbeans](#).



Share Improve this answer

answered Nov 13, 2013 at 0:50

Follow



[Florent](#)

1,389 ● 1 ● 15 ● 15



---

1 I vastly prefer cgdb now that I've gotten used to it, but Netbeans was such a great stepping stone from VS. – [zzxyz](#)  
Feb 1, 2018 at 20:16

---



3

What can be stepped through is going to be limited by the debugging information that g++ produces, to a large extent. Emacs provides an interface to gdb that lets you control it via the toolbars/menus and display data in separate windows, as well as type gdb commands directly. Eclipse's CDT provides similar tools. I've heard of Anjuta and Code::Blocks but never used them.



Share Improve this answer

answered Sep 17, 2008 at 1:47

Follow



[Allen](#)

5,090 ● 24 ● 30



As someone familiar with Visual Studio, I've looked at several open source IDE's to replace it, and KDevelop comes the closest IMO to being something that a Visual



3



C++ person can just sit down and start using. When you run the project in debugging mode, it uses gdb but kdevelop pretty much handles the whole thing so that you don't have to know it's gdb; you're just single stepping or assigning watches to variables.

It still isn't as good as the Visual Studio Debugger, unfortunately.

Share Improve this answer

answered Sep 17, 2008 at 2:16

Follow



Dennis Ferron



2



You don't mention whether you are using Windows or UNIX.

On UNIX systems, [KDevelop](#) is good but I use [KDbg](#) because it is easy to use and will also work with apps not developed in KDevelop.

[Eclipse](#) is good on both platforms.

On Windows, there is a great package called [Wascana Desktop Developer](#) which is Eclipse CDT and MinGW all packaged up and preconfigured nicely for the minimum of pain. Its the best thing I've found for developing GNU code on Windows.

I have used all these debuggers and none of them are as good as MS Dev Studio. Eclipse/Wascana is probably the closest but it does have limitations like you cannot step

into DLLs and it doesn't do as good a job at examining variables.

Share Improve this answer

answered Feb 2, 2009 at 1:01

Follow



[Adam Pierce](#)

34.3k ● 23 ● 71 ● 89



2

The [Code:Blocks](#) C++ IDE has a graphical wrapper, with a few of the features you want, but nothing like the power of VS.



Share Improve this answer

answered Nov 24, 2011 at 11:15

Follow



[crobar](#)

2,929 ● 4 ● 31 ● 47



2

[VisualGDB](#) is another Visual Studio plugin to develop and debug applications on linux and embedded platforms.



Share Improve this answer

answered May 27, 2014 at 23:08

Follow



[Sergey Zhukov](#)

1,372 ● 1 ● 13 ● 24



2

Have you ever taken a look at [DS-5 debugger](#)?

There is a paid version which includes a lot of helpful features, but you can also use Community Edition for free



(which is also quite useful especially for embedded systems).



I have a positive experience with this tool when debugging Android applications on real device using eclipse.

Share Improve this answer

edited Sep 25, 2017 at 14:01

Follow



user5248982

answered Apr 7, 2015 at 11:56



Viktor Malyi

2,386 ● 2 ● 24 ● 40

---

[Link](#) has expired. New ones [here](#). – [parthasarathy](#) Jul 14, 2020 at 5:41

---



KDevelop works pretty well.

1

Share Improve this answer

answered Sep 17, 2008 at 1:53

Follow



Harold Ekstrom

1,538 ● 8 ● 7



Have you tried gdb -w with cygwin gdb. It is supposed to have a windows interface which works fairly well.

1



The only problem I found is that on my present machine it didn't run that way until after I installed ddd. I suspect that it requires tcltk which was installed when I installed ddd.



Share Improve this answer

answered Jan 2, 2009 at 1:52

Follow



**BubbaT**

1,840 ● 4 ● 21 ● 25



Latest version of Geany supports it (only on Linux, though)

0

Share Improve this answer

answered Sep 17, 2008 at 2:08

Follow



**Milan Babuřkov**

61k ● 49 ● 130 ● 180



If you are looking for gdb under Visual Studio, then check [WinGDB](#).

0

Share Improve this answer

answered Dec 3, 2009 at 20:35

Follow



**Dominic.wig**

314 ● 3 ● 5



In the last 15 months I use insight (came with FC6). It is not great, it is written in Tcl/Tk, but it is simple and useful.

0



DDD is of similar quality / utility, but somewhat harder to use (various GUI gotchas and omissions). I also tried to integrate gdb with my IDE, SlickEdit. It worked OK (I played some 4 hours with it), but I did not like the GUI context switches. I like my IDE to remain unchanged while I am debugging; on Windows I use SlickEdit for IDE and Visual Studio Debugger for debugging. So from the 3: Insight, DDD and SlickEdit, Insight is my 1st choice, I use it >95% of the time, command-line gdb and DDD make up the other 5%. If I get the chance, I will eval Eclipse at some point, my work PC does not seem to have enough RAM (1GB only) to run Eclipse reasonably well.

I have also heard a lot of praise for TotalView, including 1st hand during a job interview. I obtained an eval for our company in late 2008, but in the end we did not proceed as gdb was good enough for our needs; and it is free and ubiquitous.

Share Improve this answer

edited Dec 3, 2009 at 21:02

Follow

answered Dec 3, 2009 at 20:53



Radim Cernej

794 ● 1 ● 10 ● 25



Use [www.zero-bugs.com/](http://www.zero-bugs.com/) Zero debugger, it requires C++0x support from gcc

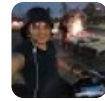
0



Share Improve this answer

answered Feb 16, 2012 at 7:21

Follow



siddhusingh

1,858 ● 4 ● 25 ● 31



---

1 [www.zero-bugs.com/](http://www.zero-bugs.com/) website is down – [vharron](#) Jul 24, 2014 at 3:12

---



0



I was searching for a debugger to step through a running programm. Say: Attach. The programm was build with eclipse, but because of maybe some multithreading obstrucles, no sourcefiles where fond. What ever.

I got very comfortable with NetBeans.



- [debug] from menu -> Attach Deugger...
- as process chose the one to debug
- as project [new project]

Now the window disappars and you see nothing. detach from the process. The Read Square "Stop" helps.

- import source from the project as e.g. folder.  
".../MyProject/src
- Now it appears in your project, and you can set breakpoints.
- again ttach debugger
- chose the process to debug.

- debugger should stop if programm reaches next breakpoint.

Going to [window] -> [Debugging] -> Will your window make comfortable.

Share Improve this answer

answered Jul 4, 2016 at 19:41

Follow



Cutton Eye

3,549 ● 3 ● 22 ● 45



**Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.