



Linux Ubuntu 26.04 Manual Pages on command 'vgdb.1'

\$ man vgdb.1

VGDB(1) vgdb VGDB(1)

NAME

vgdb - intermediary between Valgrind and GDB or a shell

SYNOPSIS

vgdb [options]

DESCRIPTION

vgdb ("Valgrind to GDB") is used as an intermediary between Valgrind and GDB or a shell. It has three usage modes:

1. As a standalone utility, it is used from a shell command line to send monitor commands to a process running under Valgrind. For this usage, the vgdb OPTION(s) must be followed by the monitor command to send. To send more than one command, separate them with the -c option.
2. In combination with GDB "target remote |" command, it is used as the relay application between GDB and the Valgrind gdbserver. For this usage, only OPTION(s) can be given, but no COMMAND can be given.
3. In the --multi mode, vgdb uses the extended remote protocol to communicate with GDB. This allows you to view output from both valgrind and GDB in the GDB session. This is accomplished via the "target extended-remote | vgdb --multi". In this mode you no longer need to start valgrind yourself. vgdb will start up valgrind when gdb tells it to run a new program. For this usage, the vgdb OPTIONS(s) can also include --valgrind and --vargs to describe how valgrind should be started.

OPTIONS

--pid=<number>

Specifies the PID of the process to which vgdb must connect to. This option is useful in case more than one Valgrind gdbserver can be connected to. If the --pid argument is not given and multiple Valgrind gdbserver processes are running, vgdb will report the list of such processes and then exit.

`--vgdb-prefix`

Must be given to both Valgrind and vgdb if you want to change the default prefix for the FIFOs (named pipes) used for communication between the Valgrind gdbserver and vgdb.

`--wait=<number>`

Instructs vgdb to search for available Valgrind gdbservers for the specified number of seconds. This makes it possible start a vgdb process before starting the Valgrind gdbserver with which you intend the vgdb to communicate. This option is useful when used in conjunction with a --vgdb-prefix that is unique to the process you want to wait for.

Also, if you use the --wait argument in the GDB "target remote" command, you must set the GDB remotetimeout to a value bigger than the --wait argument value. See option --max-invoke-ms (just below) for an example of setting the remotetimeout value.

`--max-invoke-ms=<number>`

Gives the number of milliseconds after which vgdb will force the invocation of gdbserver embedded in Valgrind. The default value is 100 milliseconds. A value of 0 disables forced invocation. The forced invocation is used when vgdb is connected to a Valgrind gdbserver, and the Valgrind process has all its threads blocked in a system call.

If you specify a large value, you might need to increase the GDB "remotetimeout" value from its default value of 2 seconds. You should ensure that the timeout (in seconds) is bigger than the --max-invoke-ms value. For example, for --max-invoke-ms=5000, the following GDB command is suitable:

```
(gdb) set remotetimeout 6
```

`--cmd-time-out=<number>`

Instructs a standalone vgdb to exit if the Valgrind gdbserver it is connected to does not process a command in the specified number of seconds. The default value is to never time out.

`--port=<portnr>`

Instructs vgdb to use tcp/ip and listen for GDB on the specified port nr rather than to use a pipe to communicate with GDB. Using tcp/ip allows one to have GDB running on one computer and debugging a Valgrind process running on another target computer. Example:

On the target computer, start your program under valgrind using

```
valgrind --vgdb-error=0 prog
```

and then in another shell, run:

```
vgdb --port=1234
```

On the computer which hosts GDB, execute the command:

```
gdb prog
```

```
(gdb) target remote targetip:1234
```

where targetip is the ip address or hostname of the target computer.

--vgdb-multi

Makes vgdb start in extended-remote mode and to wait for gdb to tell us what to run.

--valgrind

The path to valgrind to use, in extended-remote mode. If not specified, the system

valgrind will be launched.

--vargs

Options to run valgrind with, in extended-remote mode. For example -q. Everything

following --vargs will be provided as arguments to valgrind as is. This is only for

supplying valgrind options. The program executable and its arguments are specified

separately by GDB and should NOT be included here.

-c

To give more than one command to a standalone vgdb, separate the commands by an option

-c. Example:

```
vgdb v.set log_output -c leak_check any
```

-l

Instructs a standalone vgdb to report the list of the Valgrind gdbserver processes

running and then exit.

-T

Instructs vgdb to add timestamps to vgdb information messages.

-D

Instructs a standalone vgdb to show the state of the shared memory used by the Valgrind

gdbserver. vgdb will exit after having shown the Valgrind gdbserver shared memory state.

-d

Instructs vgdb to produce debugging output. Give multiple -d args to increase the

verbosity. When giving -d to a relay vgdb, you better redirect the standard error

(stderr) of vgdb to a file to avoid interaction between GDB and vgdb debugging output.

SEE ALSO

valgrind(1), `$INSTALL/share/doc/valgrind/html/index.html` or

<http://www.valgrind.org/docs/manual/index.html>,

Debugging your program using Valgrind's gdbserver and GDB[1] vgdb[2], Valgrind monitor commands[3].

AUTHOR

Philippe Waroquiers.

NOTES

1. Debugging your program using Valgrind's gdbserver and GDB

<http://www.valgrind.org/docs/manual/manual-core-adv.html#manual-core-adv.gdbserver>

2. vgdb

<http://www.valgrind.org/docs/manual/manual-core-adv.html#manual-core-adv.vgdb>

3. Valgrind monitor commands

<http://www.valgrind.org/docs/manual/manual-core-adv.html#manual-core-adv.valgrind-monitor-commands>

Release 3.26.0

02/11/2026

VGDB(1)