

NAME

lldb-server - LLDB Documentation

SYNOPSIS

lldb-server v[ersion]

lldb-server g[dbserver] [*options*]

lldb-server p[atform] [*options*]

DESCRIPTION

lldb-server provides the server counterpart of the LLVM debugger. The server runs and monitors the debugged program, while the user interfaces with it via a client, either running locally or connecting remotely.

All of the code in the LLDB project is available under the Apache 2.0 License with LLVM exceptions.

COMMANDS

The first argument to lldb-server specifies a command to run.

v[ersion]

Prints lldb-server version and exits.

g[dbserver]

Runs the server using the gdb-remote protocol. LLDB can afterwards connect to the server using *gdb-remote* command.

p[atform]

Runs the platform server. LLDB can afterwards connect to the server using *platform select*, followed by *platform connect*.

GDBSERVER COMMAND

lldb-server g[dbserver] [*options*] [[*host*]:*port*] [--] *program args...*

CONNECTION**host:port**

Specifies the hostname and TCP port to listen on. Obligatory unless another listening option is used. If host is empty, *localhost* will be used. If port is zero, a random port will be selected, and written as specified by *--pipe* or *--named-pipe* options.

--fd <fd>

Communicate over the given file descriptor instead of sockets.

--named-pipe <name>

Write the listening port number to the specified named pipe.

--pipe <fd>

Write the listening port number to the specified pipe (fd).

--reverse-connect

Connect to the client instead of passively waiting for a connection. In this case, [host]:port denotes the remote address to connect to.

GENERAL OPTIONS**--help**

Prints out the usage information and exits.

--log-channels <channel1 categories...:channel2 categories...>

Channels to log. A colon-separated list of entries. Each entry starts with a channel followed by a space-separated list of categories.

--log-file <file>

Destination file to log to. If empty, log to stderr.

--setsid

Run lldb-server in a new session.

TARGET SELECTION**--attach <pid-or-name>**

Attach to the process given by a (numeric) process id or a name.

-- program args

Launch a program for debugging.

If neither of target options are used, **lldb-server** is started without a specific target. It can be afterwards instructed by the client to launch or attach.

PLATFORM COMMAND

lldb-server p[latform] [*options*] --server --listen [[*host*]:*port*]

CONNECTION

--server

Run in server mode, handling multiple connections. If this is not specified, lldb-server will accept only one connection and exit when it is finished.

--listen <host>:<port>

Hostname and port to listen on. Obligatory. If *port* is zero, a random port will be used.

--socket-file <path>

Write the listening socket port number to the specified file.

GENERAL OPTIONS

--log-channels <channel1 categories...:channel2 categories...>

Channels to log. A colon-separated list of entries. Each entry starts with a channel followed by a space-separated list of categories.

--log-file <file>

Destination file to log to. If empty, log to stderr.

GDB-SERVER CONNECTIONS

--gdbserver-port <port>

Define a port to be used for gdb-server connections. Can be specified multiple times to allow multiple ports. Has no effect if --min-gdbserver-port and --max-gdbserver-port are specified.

--min-gdbserver-port <port>

--max-gdbserver-port <port>

Specify the range of ports that can be used for gdb-server connections. Both options need to be specified simultaneously. Overrides --gdbserver-port.

--port-offset <offset>

Add the specified offset to port numbers returned by server. This is useful if the server is running behind a firewall, and a range of ports is redirected to it with an offset.

EXAMPLES

The server can be started in several modes.

In order to launch a new process inside the debugger, pass the path to it and the arguments to the debugged executable as positional arguments. To disambiguate between arguments passed to lldb and arguments passed to the debugged executable, arguments starting with a - must be passed after --. The server will launch the new executable and stop it immediately, waiting for the client to connect.

```
lldb-server g :1234 /path/to/program program-argument -- --program-option
```

For convenience, passing the executable after -- is also supported.

```
lldb-server g :1234 -- /path/to/program program-argument --program-option
```

In order to attach to a running process, pass --attach along with the process identifier or name.

The process will be stopped immediately after starting the server. Note that terminating the server will usually cause the process to be detached and continue execution.

```
lldb-server g :1234 --attach 12345 lldb-server g :1234 --attach program-name
```

Use *gdb-remote* command to connect to the server:

```
(lldb) gdb-remote 1234
```

lldb-server can also be started without an inferior. In this case, the client can select the target after connecting to the server. Note that some commands (e.g. *target create*) will disconnect and launch a local lldb-server instead.

```
lldb-server g :1234
```

```
(lldb) gdb-remote 1234 (lldb) process launch a.out
```

SEE ALSO

The LLDB project page <https://lldb.llvm.org> has many different resources for **lldb-server** users.

AUTHOR

LLVM project

COPYRIGHT

2007-2021, The LLDB Team