



LLDB Statusline

Jonas Devlieghere

EuroLLVM 2025

zsh ~

```
jonas@jonas-mac-studio build-debug % lldb ./bin/count
(lldb) target create "./bin/count"
Current executable set to '/Users/jonas/llvm/build-debug/bin/count' (arm64).
(lldb) b main
Breakpoint 1: where = count`main + 64 at count.c:24:7, address = 0x0000000100001558
(lldb) r
Process 59024 launched: '/Users/jonas/llvm/build-debug/bin/count' (arm64)
Process 59024 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
  frame #0: 0x0000000100001558 count`main(argc=1, argv=0x000000016fdfedb8) at count.c:24:7
   21      size_t Count, NumLines, NumRead;
   22      char Buffer[4096], *End;
   23
→  24      if (argc != 2) {
   25          fprintf(stderr, "usage: %s <expected line count>\n", argv[0]);
   26          return 2;
   27      }
(lldb)
```

zsh ~

```
jonas@jonas-mac-studio build-debug % lldb ./bin/count
(lldb) target create "./bin/count"
Current executable set to '/Users/jonas/llvm/build-debug/bin/count' (arm64).
(lldb) b main
Breakpoint 1: where = count`main + 64 at count.c:24:7, address = 0x0000000100001558
(lldb) r
Process 59024 launched: '/Users/jonas/llvm/build-debug/bin/count' (arm64)
Process 59024 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1
  frame #0: 0x0000000100001558 count`main(argc=1, argv=0x000000016fdfedb8) at count.c:24:7
   21      size_t Count, NumLines, NumRead;
   22      char Buffer[4096], *End;
   23
→  24      if (argc != 2) {
   25          fprintf(stderr, "usage: %s <expected line count>\n", argv[0]);
   26          return 2;
   27      }
(lldb) settings set show-statusline true
(lldb)
```

count | count.c:24:7 | breakpoint 1.1

```
27 }  
(lldb) settings set show-statusline true  
(lldb) █
```

```
count | count.c:24:7 | breakpoint 1.1
```

```
(lldb) settings show statusline-format  
statusline-format (format-string) =
```

```
"${ansi.negative}
```

```
{ ${target.file.basename} }
```

```
{ | ${line.file.basename}:${line.number}:${line.column} }
```

```
{ | ${thread.stop-reason} }
```

```
{ | { ${progress.count} } ${progress.message} } "
```

```
27 }  
(lldb) settings set show-statusline true  
(lldb) settings set statusline-format "${ansi.bg.black}${ansi.fg.white} 🐛  ${target.file.basename}}{ | ${line.file.base  
name}:${line.number}:${line.column}}{ | ${thread.stop-reason}}{ | ${progress.count} }${progress.message}}"  
(lldb) █
```

```
🐛 count | count.c:24:7 | breakpoint 1.1
```

```
(lldb) settings show statusline-format
```

```
statusline-format (format-string) =
```

```
"${ansi.bg.black}${ansi.fg.white} 🐛
```

```
  ${target.file.basename}}  
  { | ${line.file.basename}:${line.number}:${line.column}}
```

```
  { | ${thread.stop-reason}}
```

```
  { | ${progress.count} }${progress.message}}"
```

```
27 }  
(lldb) settings set show-statusline true  
(lldb) settings set statusline-format "${ansi.bg.black}${ansi.fg.white} 🐛 {${target.file.basename}}{ | ${line.file.base  
name}:${line.number}:${line.column}}{ | ${thread.stop-reason}}{ | {${progress.count} }${progress.message}}"  
(lldb) █
```

```
🐛 count | count.c:24:7 | breakpoint 1.1
```

```
(lldb) settings show statusline-format  
statusline-format (format-string) =  
    "${ansi.bg.black}${ansi.fg.white} 🐛  
    {${target.file.basename}}  
    { | ${line.file.basename}:${line.number}:${line.column}}  
    { | ${thread.stop-reason}}  
    { | {${progress.count} }${progress.message}}"
```

```
27 }  
(lldb) settings set show-statusline true  
(lldb) settings set statusline-format "${ansi.bg.black}${ansi.fg.white} 🐛  {${target.file.basename}}{ | ${line.file.base  
name}:${line.number}:${line.column}}{ | ${thread.stop-reason}}{ | {${progress.count} }${progress.message}}"  
(lldb) █
```

```
🐛 count | count.c:24:7 | breakpoint 1.1
```

```
(lldb) settings show statusline-format  
statusline-format (format-string) =  
"${ansi.bg.black}${ansi.fg.white} 🐛  
  {${target.file.basename}}  
  { | ${line.file.basename}:${line.number}:${line.column}}  
  { | ${thread.stop-reason}}  
  { | {${progress.count} }${progress.message}}"
```

```
27 }  
(lldb) settings set show-statusline true  
(lldb) settings set statusline-format "${ansi.bg.black}${ansi.fg.white} 🐛  ${target.file.basename}}{ | ${line.file.base  
name}:${line.number}:${line.column}}{ | ${thread.stop-reason}}{ | ${progress.count} }${progress.message}}"  
(lldb) █
```

```
🐛 count | count.c:24:7 | breakpoint 1.1
```

```
(lldb) settings show statusline-format  
statusline-format (format-string) =  
    "${ansi.bg.black}${ansi.fg.white} 🐛  
    ${target.file.basename}}  
    { | ${line.file.basename}:${line.number}:${line.column}}  
    { | ${thread.stop-reason}}  
    { | ${progress.count} }${progress.message}}}"
```

```
27 }  
(lldb) settings set show-statusline true  
(lldb) settings set statusline-format "${ansi.bg.black}${ansi.fg.white} 🐛  ${target.file.basename}}{ | ${line.file.base  
name}:${line.number}:${line.column}}{ | ${thread.stop-reason}}{ | ${progress.count} }${progress.message}}"  
(lldb) add-dsym -S  
█
```

```
🐛 count | count.c:24:7 | breakpoint 1.1 | Downloading symbol file for: dyld (84E695DB-7E4B-34BF-9B0D-2C7EF0013D9E)
```

```
(lldb) settings show statusline-format  
statusline-format (format-string) =  
    "${ansi.bg.black}${ansi.fg.white}  
    ${target.file.basename}}  
    { | ${line.file.basename}:${line.number}:${line.column}}  
    { | ${thread.stop-reason}}  
    { | ${progress.count} ${progress.message}}}"
```

Format Strings

Reference on lldb.llvm.org

 LLDB

Search

USING LLDB

Tutorial

GDB to LLDB command map

Frame and Thread Format

Variable Formatting

Symbolication

Symbols on macOS

Remote Debugging

Tracing with Intel Processor Trace

On Demand Symbols

Using LLDB On AArch64 Linux

JSON Symbol File Format

Troubleshooting

Links

Man Page

SCRIPTING LLDB

Python Scripting

Python Reference

Python API

Variables

A complete list of currently supported format string variables is listed below:

Variable Name	Description
<code>file.basename</code>	The current compile unit file basename for the current frame.
<code>file.fullpath</code>	The current compile unit file fullpath for the current frame.
<code>language</code>	The current compile unit language for the current frame.
<code>frame.index</code>	The frame index (0, 1, 2, 3...)
<code>frame.no-debug</code>	Evaluates to true if the frame has no debug info.
<code>frame.pc</code>	The generic frame register for the program counter.
<code>frame.sp</code>	The generic frame register for the stack pointer.
<code>frame.fp</code>	The generic frame register for the frame pointer.
<code>frame.flags</code>	The generic frame register for the flags register.
<code>frame.reg.NAME</code>	Access to any platform specific register by name (replace <code>NAME</code> with the name of the desired register).
<code>function.name</code>	The name of the current function or symbol.
<code>function.name-with-args</code>	The name of the current function with arguments and values or the symbol name.
<code>function.name-without-args</code>	The name of the current function without arguments and values (used to include a function name in-line in the <code>disassembly-format</code>)
<code>function.mangled-name</code>	The mangled name of the current function or symbol.
<code>function.pc-offset</code>	The program counter offset within the current function or symbol
<code>function.addr-offset</code>	The offset in bytes of the current function, formatted as " + dddd"
<code>function.addr-with-args</code>	Similar to <code>function.name-with-args</code> except that there are no spaces:

ON THIS PAGE

Format Strings

Variables

Control Characters

Desensitizing Characters in the Format String

Scoping

Making the Frame Format

Making Your own Formats

Motivation

Progress Events

- Inline progress events are fragile
- Holding back new progress events

Customization

- Users like to customize lldb
- Popular request is to configure the prompt

📁 Implementation



ncurses

- Efficient and established way to create textual interfaces
- Needs to clear the screen to control it



ANSI escape codes

- How we implement colors
multiline editing
- Escape sequence to adjust the
scroll window

🔧 Future Work

Extend the format strings

- Support more format variables
- Default values
- Alignment & padding

