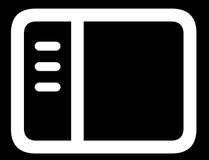
What is LLDB-DAP?

Jonas Devlieghere

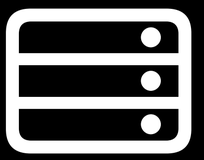
EuroLLVM 2025

Debug Adapter Protocol

Debug Adapter Protocol



client

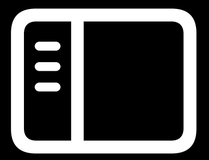


lldb-dap

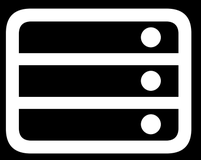
```
{
  "command": "breakpointLocations",
  "arguments": {
    "source": {
      "name": "Driver.cpp",
      "path": "/path/to/Driver.cpp"
    },
    "line": 780
  },
  "type": "request",
  "seq": 3
}
```

https://microsoft.github.io/debug-adapter-protocol/specification#Requests_BreakpointLocations

Debug Adapter Protocol



client

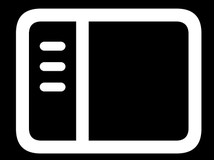


lldb-dap

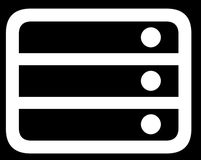
```
{
  "command": "breakpointLocations",
  "arguments": {
    "source": {
      "name": "Driver.cpp",
      "path": "/path/to/Driver.cpp"
    },
    "line": 780
  },
  "type": "request",
  "seq": 3
}
```

https://microsoft.github.io/debug-adapter-protocol/specification#Requests_BreakpointLocations

Debug Adapter Protocol



client



lldb-dap


```
{
  "body": {
    "breakpoints": [
      {
        "column": 19,
        "line": 780
      }
    ]
  },
  "command": "breakpointLocations",
  "request_seq": 3,
  "seq": 1,
  "success": true,
  "type": "response"
}
```


https://microsoft.github.io/debug-adapter-protocol/specification#Requests_BreakpointLocations

DAP in LLDB

DAP in LLDB

lldb-dap

The DAP server

LLDB-DAP

Visual Studio Code Extension

lldb-dap

The DAP server

Standalone DAP server binary

- Part of your toolchain (part of Xcode 16 & LLVM 19)
- Uses the SB API and links against `libLLDB`

Can be used with any editor supporting the protocol

- Vim, Sublime, VS Code, Emacs, etc

LLDB-DAP

The Visual Studio Code Extension

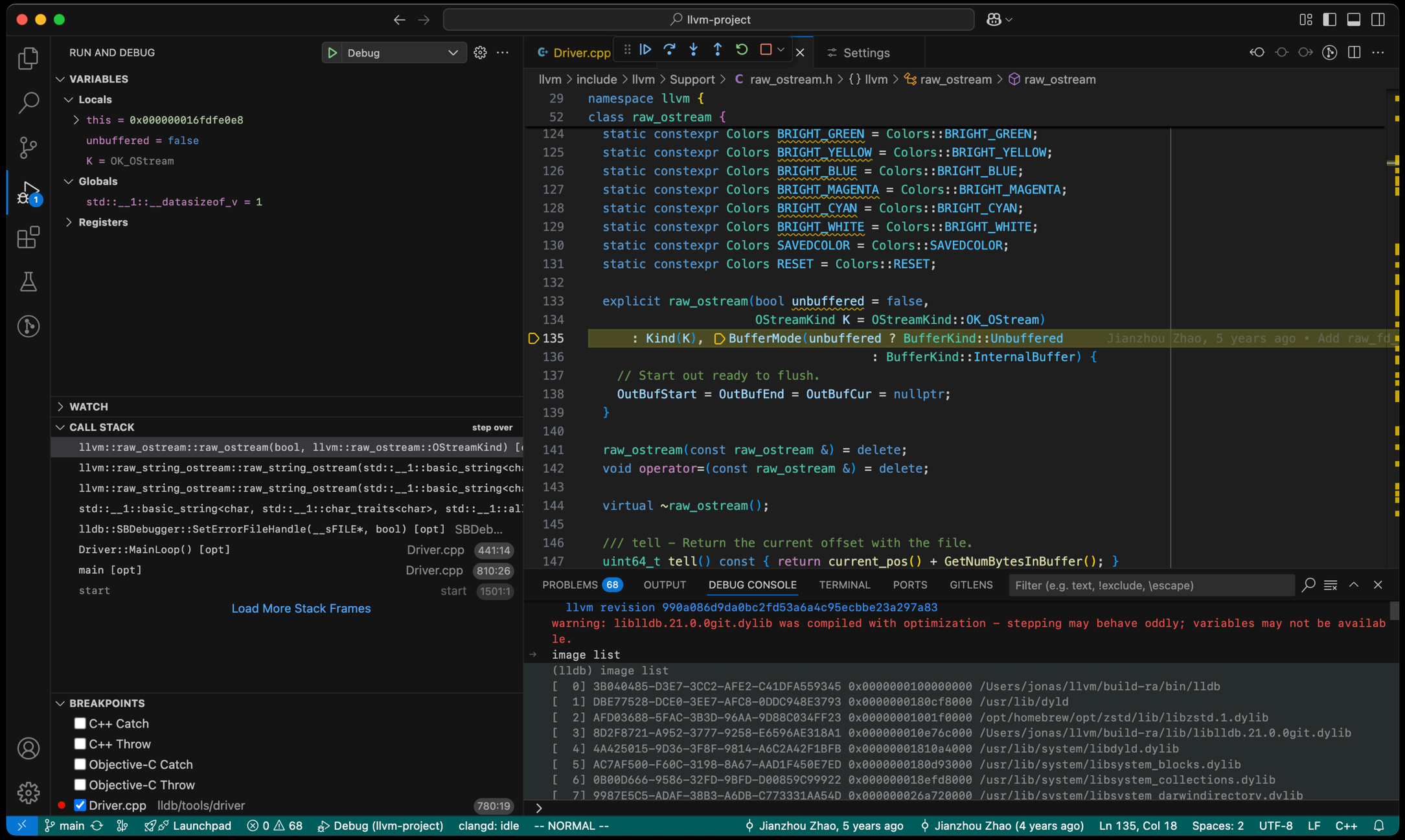
Visual Studio Code extension

- Available on the Visual Studio Marketplace
- Relies on the `lldb-dap` binary *(not included!)*

Integration into VS Code

- Settings UI
- Launch configuration templates

https://marketplace.visualstudio.com/items?itemName=llvm-vs-code-extensions.lldb-dap


llvm-project
RUN AND DEBUG
Debug
VARIABLES
Locals
this = 0x000000016fdfe0e8
unbuffered = false
K = OK_OStream
Globals
std::__1::__datasizeof_v = 1
Registers
WATCH
CALL STACK
step over
llvm::raw_ostream::raw_ostream(bool, llvm::raw_ostream::OStreamKind) [
llvm::raw_string_ostream::raw_string_ostream(std::__1::basic_string<cha
llvm::raw_string_ostream::raw_string_ostream(std::__1::basic_string<cha
std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::al
lldb::SBDebugger::SetErrorFileHandle(__sFILE*, bool) [opt] SBDeb...
Driver::MainLoop() [opt] Driver.cpp 441:14
main [opt] Driver.cpp 810:26
start start 1501:1
Load More Stack Frames
BREAKPOINTS
C++ Catch
C++ Throw
Objective-C Catch
Objective-C Throw
Driver.cpp lldb/tools/driver 780:19
Driver.cpp
Settings
llvm > include > llvm > Support > raw_ostream.h > { } llvm > raw_ostream > raw_ostream
29 namespace llvm {
52 class raw_ostream {
124 static constexpr Colors BRIGHT_GREEN = Colors::BRIGHT_GREEN;
125 static constexpr Colors BRIGHT_YELLOW = Colors::BRIGHT_YELLOW;
126 static constexpr Colors BRIGHT_BLUE = Colors::BRIGHT_BLUE;
127 static constexpr Colors BRIGHT_MAGENTA = Colors::BRIGHT_MAGENTA;
128 static constexpr Colors BRIGHT_CYAN = Colors::BRIGHT_CYAN;
129 static constexpr Colors BRIGHT_WHITE = Colors::BRIGHT_WHITE;
130 static constexpr Colors SAVEDCOLOR = Colors::SAVEDCOLOR;
131 static constexpr Colors RESET = Colors::RESET;
132
133 explicit raw_ostream(bool unbuffered = false,
134 OStreamKind K = OStreamKind::OK_OStream)
135 : Kind(K), BufferMode(unbuffered ? BufferKind::Unbuffered Jianzhou Zhao, 5 years ago • Add raw_fd
136 : BufferKind::InternalBuffer) {
137 // Start out ready to flush.
138 OutBufStart = OutBufEnd = OutBufCur = nullptr;
139 }
140
141 raw_ostream(const raw_ostream &) = delete;
142 void operator=(const raw_ostream &) = delete;
143
144 virtual ~raw_ostream();
145
146 /// tell - Return the current offset with the file.
147 uint64_t tell() const { return current_pos() + GetNumBytesInBuffer(); }
PROBLEMS 68
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Filter (e.g. text, !exclude, \escape)
llvm revision 990a086d9da0bc2fd53a6a4c95ecbbe23a297a83
warning: liblldb.21.0.0git.dylib was compiled with optimization - stepping may behave oddly; variables may not be availab
le.
image list
(lldb) image list
[0] 3B040485-D3E7-3CC2-AFE2-C41DFA559345 0x0000000100000000 /Users/jonas/llvm/build-ra/bin/lldb
[1] DBE77528-DCE0-3EE7-AFC8-0DDC948E3793 0x0000000180cf8000 /usr/lib/dyld
[2] AFD03688-5FAC-3B3D-96AA-9D88C034FF23 0x00000001001f0000 /opt/homebrew/opt/zstd/lib/libzstd.1.dylib
[3] 8D2F8721-A952-3777-9258-E6596AE318A1 0x000000010e76c000 /Users/jonas/llvm/build-ra/lib/liblldb.21.0.0git.dylib
[4] 4A425015-9D36-3F8F-9814-A6C2A42F1BFB 0x00000001810a4000 /usr/lib/system/libdyld.dylib
[5] AC7AF500-F60C-3198-8A67-AAD1F450E7ED 0x0000000180d93000 /usr/lib/system/libsystem_blocks.dylib
[6] 0B00D666-9586-32FD-9BFD-D00859C99922 0x000000018efd8000 /usr/lib/system/libsystem_collections.dylib
[7] 9987E5C5-ADAF-38B3-A6DB-C773331AA54D 0x000000026a720000 /usr/lib/system/libsystem_darwindirectory.dylib
main
Launchpad
0 ⚠ 68
Debug (llvm-project)
clangd: idle
-- NORMAL --
Jianzhou Zhao, 5 years ago
Jianzhou Zhao (4 years ago)
Ln 135, Col 18
Spaces: 2
UTF-8
LF
C++