

The LLVM Compiler Infrastructure

Site Map:

[Overview](#)
[Features](#)
[Documentation](#)
[Command Guide](#)
[FAQ](#)
[Publications](#)
[LLVM Projects](#)
[Open Projects](#)
[LLVM Users](#)
[Bug Database](#)
[LLVM Logo](#)
[Blog](#)
[Meetings](#)
[LLVM Foundation](#)

Download!

Download now: [LLVM 6.0.0](#)
[All Releases](#)
[APT Packages](#)
[Win Installer](#)

View the open-source
[license](#)

Search this Site

Useful Links

Mailing Lists:
[LLVM-announce](#)
[LLVM-dev](#)
[LLVM-bugs](#)
[LLVM-commits](#)
[LLVM-branch-commits](#)
[LLVM-testresults](#)

IRC Channel:
[irc.oftc.net #llvm](#)

Dev. Resources:
[doxygen](#) [ViewVC](#)
[Blog](#) [Bugzilla](#)
[Buildbot](#) [Green Dragon](#)
[LNT Coverage](#) [Scan-build](#)
[llvm-cov](#)

Release Emails

5.0.2: [May 2018](#)
 6.0.0: [Mar 2018](#)
 5.0.1: [Dec 2017](#)
 5.0.0: [Sep 2017](#)
 4.0.1: [Jul 2017](#)
 4.0.0: [Mar 2017](#)
 3.9.1: [Dec 2016](#)
 3.9.0: [Sep 2016](#)
 3.8.1: [July 2016](#)
 3.8.0: [Mar 2016](#)
 3.7.1: [Jan 2016](#)
 3.7.0: [Sep 2015](#)
 3.6.2: [Jul 2015](#)
 3.6.1: [May 2015](#)
 3.5.2: [Apr 2015](#)

LLVM Features

The [LLVM compiler system](#) for C and C++ includes the following:

- Front-ends for C, C++, Objective-C, Fortran, etc based on the GCC 4.2 parsers. They support the ANSI-standard C and C++ languages to the same degree that GCC supports them. Additionally, many GCC extensions are supported.
- A stable implementation of the LLVM instruction set, which serves as both the online and offline code representation, together with assembly (ASCII) and bytecode (binary) readers and writers, and a verifier.
- A powerful pass-management system that automatically sequences passes (including analysis, transformation, and code-generation passes) based on their dependences, and pipelines them for efficiency.
- A wide range of global scalar optimizations.
- A link-time interprocedural optimization framework with a rich set of analyses and transformations, including sophisticated whole-program pointer analysis, call graph construction, and support for profile-guided optimizations.
- An easily retargettable code generator, which currently supports X86, X86-64, PowerPC, PowerPC-64, ARM, Thumb, SPARC, Alpha, CellSPU, MIPS, MSP430, SystemZ, and XCore.
- A Just-In-Time (JIT) code generation system, which currently supports X86, X86-64, ARM, AArch64, Mips, SystemZ, PowerPC, and PowerPC-64.
- Support for generating DWARF debugging information.
- A C back-end useful for testing and for generating native code on targets other than the ones listed above.
- A profiling system similar to gprof.
- A test framework with a number of benchmark codes and applications.
- APIs and debugging tools to simplify rapid development of LLVM components.

Strengths of the LLVM System

1. LLVM uses a simple [low-level language](#) with strictly defined semantics.
2. It includes front-ends for [C](#) and [C++](#). Front-ends for Java, Scheme, and other languages are in development.
3. It includes an aggressive optimizer, including scalar, interprocedural, profile-driven, and some simple loop optimizations.
4. It supports a [life-long compilation model](#), including link-time, install-time, run-time, and offline optimization.
5. LLVM has full support for [accurate garbage collection](#).
6. The LLVM code generator is relatively easy to retarget, and makes use of a powerful target description language.
7. LLVM has extensive [documentation](#) and has hosted many [projects](#) of various sorts.
8. Many third-party users have claimed that LLVM is easy to work with and develop for. For example, the (now removed) Stacker front-end was written in 4 days by someone who started knowing nothing about LLVM. Additionally, LLVM has tools to make [development easier](#).
9. LLVM is under active development and is constantly being extended, enhanced and improved. See the status updates on the left bar to see the rate of development.
10. LLVM is freely available under an OSI-approved "three-clause BSD" [license](#).
11. LLVM is currently used by several commercial entities, who contribute many extensions and new features.

LLVM Audience

LLVM can be used in many different kinds of projects. You might be interested in LLVM if you are:

- A compiler researcher interested in compile-time, link-time (interprocedural), and runtime transformations for C and C++ programs.
- A virtual machine researcher/developer interested in a portable, language-

3.6: [Feb 2015](#)
3.5.1: [Jan 2015](#)
3.5: [Sep 2014](#)
3.4.2: [June 2014](#)
3.4.1: [May 2014](#)
3.4: [Jan 2014](#)
3.3: [Jun 2013](#)
3.2: [Dec 2012](#)
3.1: [May 2012](#)
3.0: [Dec 2011](#)
2.9: [Apr 2011](#)
2.8: [Oct 2010](#)
2.7: [Apr 2010](#)
2.6: [Oct 2009](#)
2.5: [Mar 2009](#)
2.4: [Nov 2008](#)
2.3: [Jun 2008](#)
2.2: [Feb 2008](#)
2.1: [Sep 2007](#)
2.0: [May 2007](#)
[Older News](#)

independent instruction set and compilation framework.

- An architecture researcher interested in compiler/hardware techniques.
- A security researcher interested in static analysis or instrumentation.
- An instructor or developer interested in a system for quick prototyping of compiler transformations.
- An end-user who wants to get better performance out of your code.

Want to Know More?

You can [browse the documentation online](#), try [LLVM in your web browser](#), or [download the source code](#).

Maintained by the
[llvm-admin team](#)