Bazel External Dependencies Roadmap

PUBLIC DOCUMENT
dslomov@google.com
2018-04-06
STATUS: Plan of record, living document

This document details the roadmap for improving external dependencies support and integration with package managers and other build systems in Bazel.

External dependencies work fall into three broad categories:

A. General improvements for WORKSPACE support in Bazel (see External Repositories: The Future for general background and discussion)
B. Allowing Bazel project to *depend* on other projects built with a different build tool and/or available through a package manager.
    a. bazel-deps is one tool that provide that support.
    b. A good analysis of challenges with C++ dependencies has been done by Envoy project.
C. Allowing other build systems to depend on Bazel project.
    a. Generally this includes features such as
        i. plugins for other build systems invoking Bazel
        ii. publishing Bazel-built artifacts so that they can be used by other build systems
        iii. generating project files for other build systems
    b. The full set of features needed for this is currently forming. *(We welcome your input on bazel-discuss@)*

2018 Q2:

- Improvements to WORKSPACE support
    - Implement diamond splitting for external dependencies
    - Implement execroot layout changes
    - Implement WORKSPACE.resolved support
    - Technical plan for recursive workspaces
- Bazel depends on 3rd-party
    - Experimental integration of bazel-deps into WORKSPACE.resolved mechanism
    - Experimental implementation of cmake / autoconf build rule (allowing Bazel to drive autoconf/make or cmake build)
- 3rd-party depends on Bazel
    - Gather requirements

2018 Q3

- Improvements to WORKSPACE support
  - Implement recursive workspaces
  - Implement "selective syncing" (bazel sync <specific workspace>)
  - Technical plan for better support of WORKSPACE.configured (tentative)
  - Technical plan for improvements of repository rule APIs and semantics (no restarts? action graphs?)
- Bazel depends on 3rd-party
  - Production-ready cmake / autoconf build rule
  - Production-ready bazel-deps integration
  - Build tool build rule for Java (maven? gradle?)
  - Community guidance for integrating 3rd party build systems and package managers
  - Technical plan for Python external deps support (tentative)
- 3rd-party depends on Bazel
  - (unknown)

2018 Q4 (tentative, depends on results of investigation/design in previous quarters)

- Improvements to WORKSPACE support
  - Improve repository rule API and semantics
- Integration with 3rd-party tools
  - Python pip support