C++ Dependency Management

from Package Consumption to Project Development

Boris Kolpackov

Code Synthesis

v1.1, September 2018

Packages

Node	(npm)	600,000
Python	(PyPI)	140,000
Rust	(cargo)	18,000
C++	(?)	?

Packages

Node	(npm)	600,000	
Python	(PyPI)	140,000	
Rust	(cargo)	18,000	(16,000)
C++	(?)	?	

What’s our Problem?

Handle Entire Project Development Lifecycle

- Creation
- Building and Testing
- Dependency Management
- Releasing and Publishing

`build2` Toolchain Overview

`b`	build system driver
`bpkg`	package manager (*consumption*)
`bdep`	project manager (*development*)
`git`	version control system

Example

Another *“Hello, World?”*

Example

Another *“Hello, World?”*

How About Something Real?

Example

Another *“Hello, World?”*

How About Something Real?

Like a UUID Library?

Creating Libraries

Born out of a Real Need

- Scratch your own itch
- Do one thing but well
- Don’t overengineer
- Have a test

Versioning

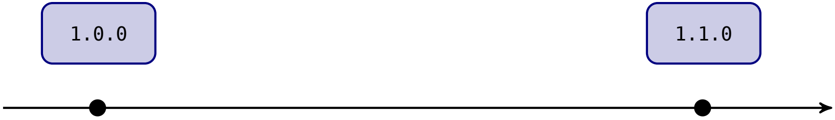
Version is a *Signaling Mechanism*

Versioning

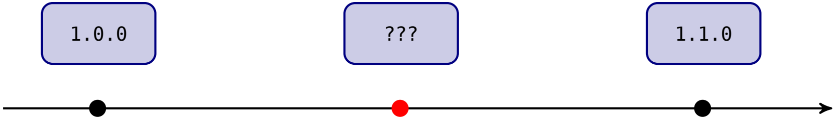
Use *SemVer* `MAJOR.MINOR.PATCH`

- Patch versions are *binary-compatible*
- ... so use `PATCH` primarily for bug fixes.
- Minor versions are *source-compatible*
- ... so use `MINOR` for additional features.
- If unsure, increment `MAJOR`.

Versioning Between Releases

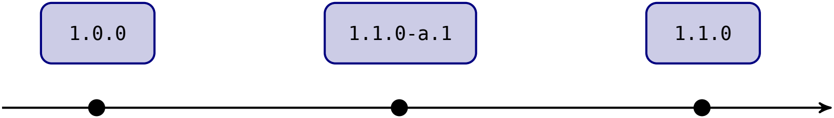


Versioning Between Releases



Versioning Between Releases

Pre-Release



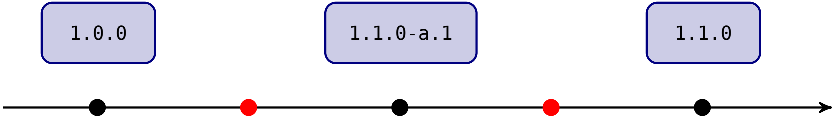
1.0.0

1.1.0-a.1

1.1.0

Versioning Between Releases

Pre-Release



Versioning Between Releases

Continuous Versioning

1.2.3	release
1.2.3-a.1	pre-release
1.2.3-b.2	pre-release
1.2.3-a.0.20180531072540.e4d2d772de35	snapshot

Package Repositories

- *Version Control*-based
- *Archive*-based

Version Control-Based Repositories

Convenient for Package Developers

- Easy to setup (GitHub)
- Project Repository == Package Repository

Archive-Based Repositories

Convenient for Package Consumers

- Searchable (packages are easy to discover)
- Reliable (packages won’t disappear)
- Secure (packages are signed)
- Fast (packages are indexed and archived)

build2 Central Repository

cppget.org

`cppget.org` Repository Policies

“Rule of One”

- `queue   --> testing` builds on one platform
- `testing --> stable` has one test
- `stable  --> legacy` fails to build on one platform

Version Constraints

`^1.2.3`	`1.X.Y`	(recommended)
`~1.2.3`	`1.2.X`	(conservative)
`>=1.2.3`		
`[1.2.3 1.2.9)`		

Summary

- `genuuid`
- `genuuid       --> libstud-uuid`
- `libstud-uuid --> cppget.org`

build2 Key Points

- Integrated build toolchain for C++
- Covers creation, development, testing, and delivery
- Uniform interface across platforms/compilers
- Archive and version control-based repositories
- Dependency-free, all you need is a C++ compiler

Offer to C++ Community

linux-gcc_7.3

linux-gcc_8.2

linux-clang_5.0

linux-clang_6.0

linux-clang_6.0_libc++

freebsd_11-clang_3.8

macos_10.12-clang_9.1

macos_10.13-clang_10.0

macos_10.13-homebrew_gcc_8.1

windows_10-msvc_14u3

windows_10-msvc_15u8

windows_10-mingw_w64_gcc_7.3

build2.org