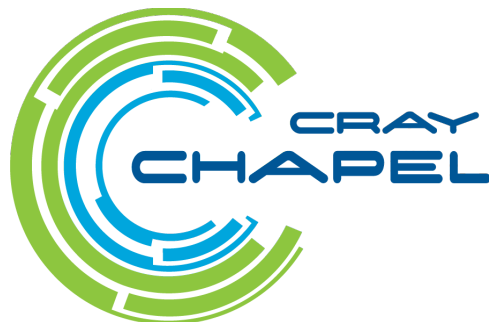




# Mason

## A Package Manager for Chapel

**Ben Albrecht**  
Chapel Team, Cray Inc.  
June 2, 2016





# Safe Harbor Statement

This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts. These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.





# Motivation for a package manager

- **Today, shared modules are checked into repository**
  - Developers must sign a CLA
  - Code must be under a compatible license
  - Code needs to be reviewed by core team
- **Modules are gated for release alongside the compiler**
- **This hinders the ability for users to contribute/share code**





- **Chapel's package manager & build tool**
- **Largely based off of Rust's Cargo**
  - Prefer to not reinvent the wheel for package management design
  - Cargo is particularly well designed
  - Rust shares many similarities with Chapel in terms of building
- **Intending to write in Chapel**
  - Begin transition to relying on Chapel for production code





# Mason: Sample Workflow

- **Initialize the project directory**

```
mason new [project name] ...
```

- **For project name foo, this produces:**

```
Foo/  
  Mason.toml  
  src/  
    Foo.chpl
```

- **Write your project code**

- **Build your project**

```
mason build
```

- This will compile Foo.chpl



# Mason: Sample Workflow

- **Dependency management**

- Add or remove dependencies

```
mason add [package] [version]
```

```
mason rm [package]
```

- Pulled in and included by `mason build`
- Dependency code is downloaded to a common pool of packages

- **Project manifest file**

- `Mason.toml`
- Tracks dependencies
  - Edited automatically by mason
  - May be edited manually
- Stores package metadata
  - Must be edited manually (name, version, authors, license, etc.)

```
[package]
name = "hello_world"
version = "0.1.0"
authors = ["Bradford Chamberlain <brad@chamberlain.com>"]
license = "Apache-2.0"

[dependencies]
Curl = "1.0.0"
```



# Mason: Package Registry

## ● Implementation

- Github repository of package manifest files
- Identical to the one in the project, plus a source url field
- Publish a package by submitting a pull request

## ● Issues

- Namespacing
  - First-come, first-served
- Versioning
  - Semantic versioning
- Integrity
  - Travis CI suite
  - Review board
- Licensing
  - SPDX
- Caching packages locally
- C dependencies

```
[package]
name = "hello_world"
version = "0.1.0"
authors = ["Brad Chamberlain <brad@chamberlain.com>"]
license = "Apache-2.0"
source = { git = "https://github.com/bradcray/hello_world", tag = "0.1.0" }

[dependencies]
Curl = "1.0.0"
```





# Mason: Implementation Details

## ● Lock file

- `Mason.lock`
- “Locks in” a build configuration from the manifest
  - Serialized DAG of all dependencies
  - Points to specific Git SHAs
- Ensures repeatable builds on other machines
- After editing a manifest, generate a new lock
  - `mason update`

## ● Syncing commands

- `mason` is a pipeline
  - `source` → `manifest` → `lock` → `dependency code`
- When `mason` commands are run, keep them in sync
  - ex. `mason add`
  - triggers `mason update`, downloaded dependencies







# Looking forward

- Pursuing mason this summer with an intern
- Looking for input & feedback from the community
- For more information, see **CHIP #9**





# Legal Disclaimer

*Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.*

*Cray Inc. may make changes to specifications and product descriptions at any time, without notice.*

*All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.*

*Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.*

*Cray uses codenames internally to identify products that are in development and not yet publically announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.*

*Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.*

*The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, and URIKA. The following are trademarks of Cray Inc.: ACE, APPRENTICE2, CHAPEL, CLUSTER CONNECT, CRAYPAT, CRAYPORT, ECOPHLEX, LIBSCI, NODEKARE, THREADSTORM. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT, and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used in this document are the property of their respective owners.*





**CRAY**  
THE SUPERCOMPUTER COMPANY