



boostpro
c o m p u t i n g

ryppi

A New Platform for (Boost) Development



Troy Straszheim

2



Why?

- We're too monolithic
- We're too imposing
- We're too slow (SVN, testing...)
- We're trying to do too many things
- Quirky tools / development practices frustrate users, packagers, and developers
- ...etc.



Why?

It's getting to be kind of a drag

Reducing Drag

- Decouple
- Lower barriers to entry
- Speed up the infrastructure
- Use (others') cool technology



Decoupling Strengthens Community



- (offered w/o proof)

What Is Ryppl?

- A package manager (APT, RPM, Emerge, Fink, Cygwin installer, AppSnap, Windows-get, ...)
- But source-based (FreeBSD ports, macports)
- And cross-platform
- A development environment
- An entry path for contribution

Technology: Pip

- Standards (backed by PEPs) for project metadata, installation process, etc.
- Project index
- Download packaged versions
- Download from VCS (git,svn,hg,bzr,...)
- Vibrant development community

Technology: Git

- Distributed Version Control System
 - Your working tree owns a repository
 - Commit changes locally
 - Push to / pull from remote repos
- Content-addressable object (file, directory, and commit) store
 - Object identity preserved across repositories
 - A commit's parentage is part of its content
- Knowledge of common ancestry plus additional merge smarts
- History rewriting

Technology: CMake

- Highly capable (e.g. installers)
- Features/support driven elsewhere
- “Traditional” architecture
 - Configure once for one target
 - Build, build, build...
- Ask David Cole

Workflows

```
$ ryppl whatever
```

I couldn't find Git in your path.
Type the path to a Git executable
here [default: I'll install one for
you]:

Install

```
$ ryppl install proj1 proj2
```

...

Would you like to run tests to make
sure this software works on your
computer (recommended)? Y/n:

Tedium Easily Avoided

```
$ ryppl install --test project1  
project2 project3
```

```
$ ryppl install --no-test project1  
project2 project3
```

Tedium Easily Avoided

```
$ ryppl install --test project1  
project2 project3  
  
$ ryppl install --no-test project1  
project2 project3  
  
$ emacs ~/.ryppl/ryppl.cfg
```



Development Workflow

```
$ ryppl checkout boost-smart_ptr
```

Development Workflow

```
$ ryppl create my-project
```

Development Workflow

```
$ ryppl publish my-project
```

How to create a public clone and
add it to the ryppl index...

Development Workflow

```
$ ryppl publish my-project
```

How to create a public clone and
add it to the ryppl index...

```
$ cd my-project
```

Development Workflow

```
$ ryppl publish my-project
```

How to create a public clone and
add it to the ryppl index...

```
$ cd my-project
```

```
my-project$ ryppl publish
```

Submitting Changes Upstream

```
$ ryppl publish other-project
```

How to create a public clone of
other-project...

Submitting Changes Upstream

```
$ ryppl publish other-project
```

How to create a public clone of
other-project...

```
$ cd other-project
```

Submitting Changes Upstream

```
$ ryppl publish other-project
```

How to create a public clone of
other-project...

```
$ cd other-project
```

```
other-project$ ryppl submit-patches
```

Submitting Changes Upstream

```
$ ryppl publish other-project
```

How to create a public clone of
other-project...

```
$ cd other-project
```

```
other-project$ ryppl submit-patches
```

Local Testing

```
$ ryppl test boost-python boost-mpl  
$ cd my-project  
my-project$ ryppl test  
my-project$ ryppl test --deep
```

Remote Testing

```
my-project$ ryppl test --remote  
my-project$ ryppl test --remote=beman-mac
```

Releasability

```
$ ryppl show-release-criteria someproj
```

What is Boost, Then?

- Vetting / Review
- Certification
 - Individual Libraries
 - Aggregate Releases
- Best Practices
- Community

What Isn't Boost?

- Tools development

What's Optional?

- Single Website
- Monolithic mailing list
- Monolithic issue tracker
- Services Hosting
- ...