

Ideas on a waf-based software development system

BV

[2014-03-25 Tue 20:18]

Random spew of thoughts.

1 Think about some examples

The package **A** produces an application. **A** requires a library and headers from package **B**. In building **A** the system must be able to locate `libB` and `B.h` output by package **B**. Locating these output files obviously depends on where they are placed by the system.

A suite of **N** packages are needed. They provide some number of applications, headers and libraries. Users of this suite must be able to run the applications. Users will extend this suite with their own packages that contain code that builds against the headers and libraries.

Of the **N** packages, **M** need development. The **N-M** packages that have been installed should be reused.

Two ways to place things:

root version the Unix filesystem approach. Have a central root directory and install all output results together. `/root/bin`, `/root/lib`, `/root/include`, etc.

ver

2 Requirements

The system must satisfy some basic requirements.

- Support multiple developers
- Leverage a central release installation

- Allow portions of the central release to be overridden by individuals with newer versions
- Manage dependencies
- Manage but minimize user environment variables
- Encourage definitions of units of code ("packages")

3 What does the system cover

- Define a unit of source code called a package which produce some derived output
- Rules for producing output are simple to express simple things, allow for complex things
- Express dependencies between packages allowing one to locate the products of another.

4 Not

This system is in the vein of

- CMT

5 Concepts

5.1 CMT

In CMT there are projects of packages. Packages can depend on other packages to form a dependency tree. Commands can be "broadcast" to packages on this tree. This broadcast is serial (but there is a parallel variant) and progresses from the leaves of the dependency sub-tree that is rooted in the "current" package. Current package is determined by the directory in which the command was broad cast.

5.2 A package

A package is a directory with a build system control file.