LCFG Buildtools Project

Stephen Quinney <squinney@inf.ed.ac.uk>

Why do we need build tools?

- We have lots of components and other locally authored software. We want:
  - Reduction of effort
  - Replication of work should be minimised
  - Focus on writing code not building packages
  - Must be easy to build packages for any platform (both for us and external packagers)

Why do we need **new** build tools?

- Code is now overgrown and unkempt
- Hard to comprehend and maintain
- Hard to extend to new platforms
- Lots of RPM specific stuff embedded
- Pretty much tied to CVS

What do the LCFG buildtools do?

- Version control
- System detection
- Configuration/Translation of files
- Building
- Packaging
- Testing

Requirement: Version Control

- It must be possible to use any version control system of choice (or none).
  - Currently we are tied to CVS (**yuck!**)
  - Makes it difficult for external users to rebuild
  - Makes it hard for external users to create new components
- Must be decoupled from build system.

Requirement: Target Platforms

- It should be possible to build a package on any Unix platform with minimal effort.
- Must be possible for other users to extend the system.
- All these should be possible:
  - Linux (Fedora, Debian, Gentoo, etc..)
  - MacOSX
  - Solaris
  - BSD

What are the options?

- Fully self-contained source distribution
  - Autotools (autoconf, automake)
  - Home-brew (the current way)
- Require “special” tools
  - CMake, SCons
  - Home-brew

Current System

- Based around a set of Makefiles:
  - buildtools.mk
  - lcfg.mk
  - os.mk
  - site.mk
  - mmaker.mk
  - test.mk
  - config.mk

What do we already require?

- GNU make
- Perl
- Bash shell
- GNU sed
- tar
- awk
- etc...

Package meta-data

- Currently in `config.mk`
- Need a better way to drive the system:
  - API for Perl code (`LCFG::Build::PkgSpec`)
  - Command line tool to query and modify (`lcfg-pkgcfg`)
- Move to `lcfg.yml` (YAML format)
- Need a converter from old to new:
  - `lcfg-cfg2meta`

Meta-data - lcfg.yml

```
abstract: LCFG Website Builder
author: Stephen Quinney <squinney@inf.ed.ac.uk>
base: lcfg
date: 28/02/08 12:28
group: LCFG
license: GPL
name: lcfgweb
platforms:
  - Fedora5
  - Fedora6
release: 2
schema: 1
translate:
  - "*.cin"
  - specfile
vcs:
  type: CVS
vendor: University of Edinburgh
version: 1.0.1
```

lcfg-pkgcfg

```
lcfg-pkgcfg --get=name

lcfg-pkgcfg --get=name --in ~/cvs/lcfg-foo

lcfg-pkgcfg --get=name --in ~/cvs/lcfg-foo/lcfg.yml

lcfg-pkgcfg --set schema=2 --set license=gpl

lcfg-pkgcfg --skeleton

lcfg-pkgcfg --skeleton --set name=foo --set version=1.0.0

lcfg-pkgcfg --in META.yml --out lcfg.yml
```

Release Tools

- Split off from the LCFG build tools suite
  - Provide standard interface to any VCS
  - `LCFG::Build::VCS`
  - `Lcfg-reltool`
- Driven by vcs info in lcfg.yml, options include:
  - Control over which VCS to use
  - ChangeLog generation
  - Checking for uncommitted files

lcfg-reltool

```
lcfg-reltool --release ~/cvs/lcfg-foo

lcfg-reltool --minorversion --quiet

lcfg-reltool --checkcommitted

lcfg-reltool --genchangelog

lcfg-reltool --no-checkcommitted --majorversion
```

Package Building

- Most packaging systems expect:
  - tar file of source
  - Package specification.
- **Aim**: Separate out generation of source from generation of package.

Typical build process

1) Add cool new feature
2) Commit to VCS and tag version
3) Export source from VCS
4) Generate source tar file from export
5) Build packages for platforms from tar file (probably using build farm).

Build Farm

- Coming Soon!
- Only for RPM based systems for now
  - Fedora Core 5, 6 (32 and 64 bit)
  - Scientific Linux 5 (32 and 64 bit)
- Based on standard Redhat technology
  - mock
  - koji