El-Get

`M-x apt-get`

Dimitri Fontaine dim@tapoueh.org

March, 30 2013

Dimitri Fontaine

2ndQuadrant France
PostgreSQL Major Contributor
Emacs Lisp Hacker

- `pgloader`, `prefix`, `skytools`, `debian`, ...
- `CREATE EXTENSION`
- `CREATE EVENT TRIGGER`
- *El-Get*
- *Emacs Lisp, Common Lisp*

Emacs: Emacs Lisp

How I used to do things

```
C-x C-/ runs the command goto-last-change
```

- Talk about it on IRC
- Search for it in `http://emacswiki.org`
- `wget` the file
- Somewhere in your `load-path`
- Beware of whole directories vs single file
- Read installation notes, edit your setup to `require` it
- Look for docs (`.info`) and install those yourself

How I used to do things

```
C-x C-/ runs the command goto-last-change
```

- Now try it!
- If you like it, you're about done (*customizing*)
- Just think about upgrading to a newer version, if needed
- (e.g. to support a newer Emacs)
- If you don't like it, undo all the previous bits

Managing Emacs Extensions

Installing is but the first step

- Version Control as if it were yours
- Sharing to other machines

How much can you install and manage this way?

You tend to only install *trusted* extensions

El-Get

Think apt-get for Emacs

Social Package Management System

- `M-x el-get-list-packages`
- `M-x el-get-describe`
- `M-x el-get-install`
- `M-x el-get-update`, `M-x el-get-reload`
- `M-x el-get-remove`
- `M-x el-get-self-update`

Features

moar please

- Package dependencies
- Cleanup command (*autoremove*)
- Extensive Info Documentation, with a Glossary!
- Stable branch is very stable...
- Easy install
- Good default setup, more complex setup supported
- Easy per-package setup (`el-get-user-package-directory`)
- Autoloads

Features

moar please

- Checksum support
- git: Shallow clones
- git: Submodules
- Per `system-type` build commands
- Windows support (users called) (yes they still exist)
- Build commands, `byte-compile`
- Recipe Checker

Benefits

What happens at `el-get-install` time?

- El-Get knows how to fetch the extension
- Even if it's more than a single file
- The *recipe* contains the necessary installation bits
- Docs get installed and integrated into your `C-h i`
- The package is *byte-compiled*
- Then `autoloaded`
- If the *recipe* says so, features are `required`

Setup

Hey it's Emacs, I want to do it my own way!

- Initializing a package involves custom actions
- `el-get-sources`
- `:before`, `:after`, `:features`
- `el-get-user-package-directory`, `init-pkgname.el`
- `M-x el-get-init`

Recipes

Gimme your recipes

All your recipes are belong to us!

- recipes are packaging script
- can be pretty simple
- can be pretty involved
- easy enough to make your own locally
- `el-get-sources` or in a local file
- require no edits of upstream sources
- find a recipe with `M-x el-get-find-recipe-file`

Recipe Example 1, el-get

el-get is self-hosted and has a *scratch* installer

```
(:name el-get
 :website "https://github.com/dimitri/el-get#readme"
 :description "Manage the external elisp bits and pieces you
 :type github
 :branch "4.stable"
 :pkgname "dimitri/el-get"
 :info    "."
 :load    "el-get.el")
```

Recipe Example 2, escreen

Escreen is available on some website, not ELPA ready, Copyright 1992, 94, 95, 97, 2001, 2005 Noah S. Friedman

```
(:name escreen
 :description "Emacs window session manager"
 :type http
 :url "http://www.splode.com/~friedman/software/emacs-lisp/sr
 :post-init (lambda ()
              (autoload ’escreen-install "escreen" nil t)))
```

Recipe Example 3

Solarized is a recent color-theme, and has been a fast moving target, hosted at github, not available as an ELPA package.

```
(:name color-theme-solarized
 :description "Emacs highlighting using Ethan Schoonover’s Sol
 :type github
 :pkgname "sellout/emacs-color-theme-solarized"
 :depends color-theme
 :prepare (progn
            (add-to-list ’custom-theme-load-path default-dire
            (autoload ’color-theme-solarized-light "color-them
              "color-theme: solarized-light" t)
            (autoload ’color-theme-solarized-dark "color-theme
              "color-theme: solarized-dark" t)))
```

Sharing Recipes

Distributed packaging: the Emacs Packages Social Network

Just email your recipe to your friend

Sharing Recipes

Distributed packaging: the Emacs Packages Social Network

- Once tested, send them to me (github, mail)
- Once pushed, `el-get-self-update`
- Available in `M-x el-get-install`
- and in `M-x el-get-describe`
- and in `M-x el-get-list-packages`

Where to get the packages from?

El-Get has more than one *method* to fetch your package

- git (with branch support), git-svn
- github, github-tar, github-zip
- http, http-tar, http-zip, ftp
- emacswiki, `M-x el-get-emacswiki-refresh`
- emacsmirror
- svn, bzr, darcs, cvs, hg, fossil
- apt-get, pacman, fink, brew
- elpa (multi repositories, `:repo`)

Installing and Using El-Get

The *scratch* installer

```
;; So the idea is that you copy/paste this code into your *sc
;; hit C-j, and you have a working el-get.
(url-retrieve
 "https://github.com/dimitri/el-get/raw/master/el-get-install
 (lambda (s)
   (end-of-buffer)
   (eval-print-last-sexp)))
```

The initial setup in .emacs

```
(add-to-list 'load-path "~/.emacs.d/el-get/el-get")

(unless (require 'el-get nil 'noerror)
  (with-current-buffer
      (url-retrieve-synchronously
       "https://raw.github.com/dimitri/el-get/master/el-get-i
    (goto-char (point-max))
    (eval-print-last-sexp)))

(el-get 'sync)
```

Active Community

Some numbers, because we know you like them

- 181 contributors
- 748 recipes included
- 1795 emacswiki files supported
- 3054+ emacsmirror repositories
- 7200+ lines of emacs-lisp fun (including 20% of comments)
- 2+ years old, long time stable, production ready

El-Get. Any question?