Writing packages for Octave

Carnë Draug
David Miguel Susano Pinto

September 21, 2015

```
my_functions
    ├── estimate_psf.m
    ├── get_otf_order.m
    ├── reconstruct_sim.m
    ├── timeit.m
    └── write_hyper_tiff.m
```

How they can package it

```
my_package
    |___ inst
    |        |___ estimate_psf.m
    |        |___ get_otf_order.m
    |        |___ reconstruct_sim.m
    |        |___ timeit.m
    |        |___ write_hyper_tiff.m
    |___ COPYING
    |___ DESCRIPTION
```

The DESCRIPTION file

```
Name: my_package
Version: 0.0.1
Date: 2015-09-21
Title: My Very Own Package
Author: Old Me
Maintainer: Current Me
License: GPLv3+
Description: A package for my octave
 functions.
Categories: SIM data
```

- An alternative to "Categories" is the INDEX file.
- "License" is optional because the COPYING file is required.

The INDEX file

```
my_package >> My Very Own Package
Structural Illumination Microscopy
  reconstruct_sim
  get_otf_order
  estimate_psf
Performance
  timeit
Image IO
  write_hyper_tiff
```

- This is used for "pkg describe -verbose pkg_name" as well as in Octave Forge online documentation.

- Allows to omit private functions or add extra functions.

```
my_package
    |
    |___ inst
    |       |
    |       |___ ...
    |
    |___ ChangeLog
    |
    |___ CITATION .......................via citation()
    |
    |___ COPYING
    |
    |___ DESCRIPTION
    |
    |___ INDEX
    |
    |___ NEWS ................................via news()
```

```
my_package
    inst
        ...
    src
        __sim__.cc
        decon_4d.cc
        Makefile
    ...
```

```
MKOCTFILE ?= mkoctfile
OCT_FILES = decon_4d.oct \
                __sim__.oct


all: ${OCT_FILES}


%.oct: %.cc
    $(MKOCTFILE) $<
```

'`pkg install`' runs '`make all`' and then copies all the
generated oct files.

configure

```
my_package

    inst

        ...

    src

        __sim__.cc

        configure.ac

        configure

        decon_4d.cc

        Makefile

    ...
```

It will also run `configure` before `make`, if there is a configure script.

Dependencies

- The DESCRIPTION file allows to list dependencies on octave and packages versions via `Depends`.
- Other dependencies in DESCRIPTION are not enforced.

```
Depends: octave (>= 3.8.0), image (>= 2.2.0)
BuildRequires: libmrc
SystemRequirements: ImageJ
```
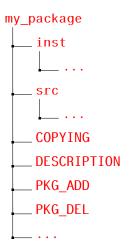
PKG_ADD and PKG_DEL

```
my_package
    |
    |___ inst
    |    |
    |    |___ ...
    |
    |___ src
    |    |
    |    |___ ...
    |
    |___ COPYING
    |
    |___ DESCRIPTION
    |
    |___ PKG_ADD
    |
    |___ PKG_DEL
    |
    |___ ...
```

Useful for:

- checking dependencies at load time
- autoload functions
- setting options, register colormaps, image readers, . . .

Caveats:

- these are scripts so be careful about leaving the userspace intact
- do not forget PKG_DEL

It's often easier to have these as PKG_ADD and PKG_DEL directives in the source files, than maintaining an actual PKG_ADD file.

A reminder

Packages exists in 3 different states:

- development. Whatever form or shape or order the files are in the repository.
- package. The structure inside the tarball, it's the only thing that is a requirement.
- installed. Once it is installed, you should not have to care where and how the files are placed and organized.
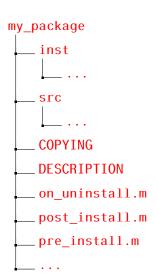
Install hooks

```
my_package
    inst
        ...
    src
        ...
    COPYING
    DESCRIPTION
    on_uninstall.m
    post_install.m
    pre_install.m
    ...
```

- called as functions, not scripts
- never saw them being used for anything other than workaround pkg bugs

The simplest
case

Adding
compiled
languages

More power

Shooting
yourself in the
foot

**Releasing
packages**

Octave Forge

Big requirements:

- GPL compatible license, including non-packaged dependencies
- up to date clone of the repository hosted in Octave Forge at release time

Advantages:

- push access to all packages for all forgers
- bug tracker shared with octave core
- packages installable via pkg's --forge option
- online documentation side-by-side with all the others

How's your package?