

Packaging guide

Sergey Basalaev

June 23, 2013

Contents

1	Introduction	1
2	Package metadata	2
3	Building using make	2
4	Examples	3
5	How to put your package in the Software Archive	4

1 Introduction

Software for Alchemy OS comes in *packages* which are essentially archives containing program files and some descriptive metadata. A special program called *package manager* is used to install, remove and upgrade packages. Alchemy OS has central web-archive of packages from which the software can be downloaded and/or upgraded in automated way.

To build a package yourself you need `pkgbuild` utility. You can install `pkgbuild` from the graphical package manager or by executing the following command in the terminal:

```
pkg install pkgbuild
```

If you cannot access internet on your phone then download the package and all its dependencies from the [software archive](#) and install them manually using

```
pkg-arh install yourfile.pkg
```

Syntax of the utility is the simplest:

```
pkgbuild      # Create binary package
pkgbuild -s   # Create source package
pkgbuild -c   # Clean project from building leftovers
```

2 Package metadata

Suppose we have a program named `foo`. All we need to do is to create a file with `.package` extension (e.g. `foo.package`) and fill it with information.

```
# First comes info about sources
Source: foo
Version: 1.0
Section: utils
Author: John Doe <jdoe@example.com>

# Second paragraph describes binary package
Package: foo
Summary: the great foo program
Depends: ${libdeps}, zip
Files:
    foo /bin
    foo.png /res/icons
    foo.desktop /res/apps
```

Here, "Files" field tells what and where will be installed. The word `${libdeps}` will be automatically substituted by the names of the libraries which are needed for the program.

Now, to build binary package simply call

```
pkgbuild
```

This will create file `foo_1.0.pkg` above your current directory. Your package is ready. You may notice that there is now `PKGBUILD` folder in your current directory. This is the contents of the binary package, you may verify that it contains all needed files in right places. Now, try to create source package:

```
pkgbuild -s
```

This will remove `PKGBUILD` folder and create file `foo_1.0.zip` above your current directory. This is just an archive with the contents of the current directory.

3 Building using make

If your project uses `make` for building, `pkgbuild` may also call it during builds. For that, add to the source paragraph of the `.package` file:

```
Build-Depends: make
```

The Makefile used by `pkgbuild` must contain at least three targets:

The first target (regardless of name) builds the whole project.

install installs program on the system. For pkgbuild to work it must install under `${DESTDIR}` instead of root.

clean removes all built files and makes directory suitable for building source package.

When `pkgbuild` is called with `-s` or `-c` option, it calls "*make clean*". When called without arguments, it successively calls "*make*", "*make install*".

4 Examples

The following two examples are existing projects in Alchemy OS. You can download them from the [package archive](#) and examine their sources.

‘hello’ program

Makefile used for building:

```
hello: hello.e
    ex hello.e -o hello

install:
    install hello ${DESTDIR}/bin

clean:
    rm -f hello
```

File `hello.package` with package information:

```
Source: hello
Section: misc
Version: 2
Author: Sergey Basalaev <sbasalaev@gmail.com>
License: BSD-2
Copyright: (c) 2011-2013 Sergey Basalaev
Build-Depends: make

Package: hello
Depends: ${libdeps}
Summary: an example of program
```

‘libexample’ library

Makefile used for building:

```
all: libexample.1.so libexample.so

libexample.1.so: example.e
    ex example.e -o libexample.1.so -slibexample.1.so

libexample.so:
    echo '#=libexample.1.so' > libexample.so

clean:
    rm -f libexample.1.so libexample.so

install: all
    install libexample.1.so libexample.so ${DESTDIR}/lib
    install example.eh ${DESTDIR}/inc
```

File libexample.package with package information.

```
Source: libexample
Section: libs
Version: 1
Author: Sergey Basalaev <sbasalaev@gmail.com>
Copyright: (c) 2013 Sergey Basalaev
License: BSD-2
Build-Depends: make

Package: libexample1
Summary: an example of a library
Depends: ${libdeps}
Files:
    /lib/libexample.1.so

Package: libexample-dev
Section: libdevel
Summary: an example of a library - development headers
Depends: libexample1
Files:
    /lib/libexample.so
    /inc
```

5 How to put your package in the Software Archive

To be included in the Software Archive the package must meet the requirements of the [Alchemy OS Policy](#). To propose a package for inclusion in the Software

Archive either send it to our [mailing list](#) or create a [package proposal](#) and attach your package and/or sources to it. It is recommended that before upload the package is also checked for common errors by `pkglint` utility.

Copyright (c) 2012–2013, Sergey Basalaev

This manual is free software; you may redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3, or (at your option) any later version.

This is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

A copy of the GNU General Public License is available on the World Wide Web (see <http://www.gnu.org/copyleft/gpl.html>). You can also obtain it by writing to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.