How to make an OSX taskwarrior package

0. Philosophy

Only Fredde tags releases. We only make builds from tagged commits. We only release builds that build cleanly without errors or warnings. We only release builds that pass 100% of the unit tests.

1. Prerequisites

- You will need an Intel Mac, running OSX 10.5 or later.
- You will need to install the Developer Tools, which are found on your OSX DVD.
- You will need git installed, version 1.5 or later. See http://git-scm.com
- · You will need autotools installed. See http://www.gnu.org/software/autoconf

2. Get the code

2.1 Clone the taskwarrior git repository. It is important that this is a throwaway clone of the repository, because we will do (locally) destructive things to it.

```
$ git clone git://tasktools.org/task.git ~/task-package.git
...
$ cd ~/task-package.git
```

2.2 Making sure you have the right version of the code. This assumes you are building taskwarrior 1.9.2, but any version number is interchangeable. Check out the correct branch, and make sure it is sitting at the correct commit, via a tag. Note that while 1.9.2 is a branch name, v1.9.2 is a tag name.

```
$ git checkout 1.9.2
$ git reset --hard v1.9.2
```

If there is an error in this step, stop immediately, capture the output, and report the errors.

3. Build taskwarrior

3.1 First build the task binary. Note the "-j 2" tells make to use both cores in your dualcore Intel CPU, which means faster compiles. If you own a quad core Mac, use "-j 4". If you own a single core Mac, just type "make".

```
$ autoreconf -f
$ ./configure
...
```

If any errors are reported, stop immediately, capture the output, and report the errors.

```
$ make -j 2
```

If there are any errors, or there are any warnings generated by the compiler stop immediately, capture the output, and report the problem. You'll need to watch as it builds.

4. Build the test suite

4.1 The test suite exists to prove that we do not break taskwarrior features from one release to the next. While this is not a perfect solution, it has saved us many times from releasing code that is inferior.

The first step is to modify the test suite Makefile to remove the Lua line. This is because we do not yet have dynamic detection of the Lua library for the unit tests.

```
$ cd ~/task-package.git/src/tests
$ vi Makefile
```

Any text editor will do, but look for this line (line 5):

LFLAGS = -L/usr/local/lib -lncurses -llua

and change it to:

LFLAGS = -L/usr/local/lib -lncurses

Now build the unit tests:

\$ make -j 2

4.2 Run all the unit tests.

```
$ ./run_all
Skipping benchmarks
Pass: 4241
Fail: 0
Skipped: 0
Runtime: 86
```

The output should look something like this, with 0 failed, and 0 skipped tests. If there are any failures or skips, stop and mailed the log file, named 'all.log' to Paul & Fredde.

5. Assemble the parts

5.1 There is a script that copies all the necessary files (binary, man pages etc) in the right place, ready for packaging. Run this:

```
$ cd ~/task-package.git/package-config/osx
$ ./update
```

6. Adjust the package details

6.1 Launch the package manager.

```
$ open -a /Developer/Applications/Utilities/PackageManager.app
```

6.2 Close the 'Untitled' window that opens - we will be using a different file. Note that PackageManager is still running - it just has no windows.

6.3 Using the File -> Open menu, open the file

/Users/<your-account>/task-package.git/package-config/osx/task.pmdoc



This is the file from the last time a package was created. It needs some adjustments. Start by clicking on the "Taskwarrior x.x.x Distribution" with a package icon in the top left part of the window.

6.4 Click on the "Configuration" button/tab.

- Change the "Title" to "Taskwarrior 1.9.2"
- Change the "Description" to "Taskwarrior 1.9.2 install for Snow Leopard"



6.5 Click on "Edit Interface ... "

There are 5 radio buttons on the left hand side - we will visit each and make changes. Click on "Background", make sure it matches the figure.



Click on "Introduction", make sure it matches the figure.

000	🗑 task.pmdoc – Interface Editor – en	
	Welcome to the Task 1.9.2 Installer	Welcome Panel
Background		 Default
Introduction	 Select "Default" to show the default welcome: 	C Embedded
🖯 Read Me	You will be guided through the steps necessary to install this software.	O File:
e License	 Select "Embedded" to enter a text to be 	Q.
⊖ Finish Up	embedded in the distribution. • Or specify your custom welcome file by dropping it here.	Different localizations cannot mix embedded and file-based resources for the same panel.
	en + - Go Back Continue	

Click on "Read Me", and on the right hand side, under "Read Me Panel", click on "File" and select the file:

/Users/<your-account>/task-package.git/package-config/osx/README.txt

000	💿 task.pmdoc – Interface Editor – en	
	Important Information	Read Me Panel
 Background Introduction Read Me License Finish Up 	 Thank you for taking a look at task! Task is a GTD, todo list, task management, command line utility with a multitude of features. It is a pontable, well supported, very active project, and it is Open Source. Task has binary distributions, online documentation, demonstration movies, and you'll find all the details at: http://taskwarrior.org At the site you'll find a wiki, discussion forums, downloads, news and more. Your contributions are especially welcome. Whether it comes in the form of code patches, ideas, discussion, bug reports, encouragement or criticism, your input is needed. Please send your support questions and code patches to: en the Co Back Continue 	None Embedded File: DSX/README.txt

Click on "License", and on the right hand side, under "License Panel", click on "File" and select the file:

/Users/ <vo< th=""><th>our-account></th><th>/task-package</th><th>.git/package</th><th>e-config/osx/</th><th>/COPYING.txt</th></vo<>	our-account>	/task-package	.git/package	e-config/osx/	/COPYING.txt
,	ar accounter,	cabin pacinage	- <u>j - o, paonaj</u>		



Click on "Finish Up", and make sure it matches the figure.



Close the Interface Editor window (it saves automatically).

6.6 Back in the "task.pmdoc" window, click on the "Requirements" tab/button, then click on the "+" button, and add a requirement rule as shown in the figure. The minimum system should be 10.5.0 or 10.6.0, depending on which package is being built.

000	i task.pmdoc
Build Build and Run	Edit Interfac
Task 1.9.2 Distribution	Task 1.9.2 (Distribution)
Contents	Configuration Requirements Actions
🕨 🖯 local	
	Required Pass if Description
	Required true Target OS Version (e.g. 10.x.x) is greater than or equal to '10.6.0'
+ \$-	

6.7 Click on the "Actions" tab/button and make sure it matches the figure.

000	e task.pmdoc	
Build Build and Run		Edit Interface
Task 1.9.2 Distribution Contents © local	Task 1.9.2 (Distribution) Configuration Requirements Actions Preinstall Actions	
+ &-		

6.8 Click on the triangle next to the "local" item in the "Contents" vertical bar on the left to expand and show a folder called "local", with the path "/usr/local", but click on the "local" next to the blue radio button, not the one next to the folder. Then click on the "Configuration" tab/button.

- Change the "Choice Name" to "local"
- · Leave the "Identifier" alone it is automatic
- Make sure "Selected" and "Enabled" are checked, but "Hidden" is not

000	
Build Build and Run	Edit Interface
Task 1.9.2 Distribution Contents	Choice: local Configuration Requirements
local /usr/local	Choice Name: local Identifier: choice 37 Initial State: Selected Enabled Hidden Destination:
	Tooltip: Description:
+ \$-	

6.9 Click on the "Requirements" tab/button and make sure it is all blank.

Build Build and Run	task.pmdoc	Edit Interface
Task 1.9.2 Distribution Contents	Choice: local Configuration Requirements	
local /usr/local	Description	
+ ö.		

6.10 Click on the blue folder on the left, which is labelled "local", and has a path of "/usr/ local". Click on the "Configuration" tab/button.

- Change the "Install" to /Users/<your-account>/task-package.git/package-config/osx/ local
- Make sure the "Destination" is "/usr/local"
- Make sure "Allow custom location" is checked
- Make sure the "Package Identifier is "com.beckingham.task192.local.pkg"
- Make sure the "Package Version" is 1.0. If you needed to make a subsequent OSX package, *for the same version of taskwarrior*, then this number would be increased to show OSX that this package supersedes the earlier one
- Make sure "Restart Action" is "None"
- Make sure "Require admin authentication" is checked
- Make sure "Package Location" is "Self-Contained"

000	🥃 task.pmdoc 💭
Build Build and Run	Edit Interface
Task 1.9.2 Distribution	Package: local
Contents	Configuration Contents Components Scripts
local /usr/local	Install: /Users/paul/task-package.git/package- *-
	Destination: /usr/local Allow custom location
	Patch: O No patch root selected
	Package Identifier: com.beckingham.task192.local.pkg
	Package Version: 1.0
	Restart Action: None
	Require admin authentication
	Packages can be stored inside your distribution/metapackage or at a URL.
	Package Location: Self-Contained *-
+ 🔷 -	

6.11 Click on the "Contents" tab/button.

- Click on all the triangles in the "local" folder to expand all directories
 Make sure "Include root in package is not checked"
- Click on the "Apply Recommendations" button

Make sure it matches the figure.

ld Build and Run				Edit Interf
Task 1.9.2	Package	e local	_	
Distribution				_
local	Configuration Contents	Components	Scripts]
			_	
	Path	Owner	Group	Mode
		paul ÷	stan ubool	+ drwxr-xr-x
		root +	wheel	+ -rwyr-yr-y
		root +	wheel	+ drwxr-xr-x
		root ‡	wheel	t drwxr-xr-x
		root ‡	wheel	t drwxr-xr-x
		root ±	wheel	* -rw-rr
		root ±	wheel	* -rw-rr
		root ±	wheel	* -rw-rr
	M NEWS	root ±	wheel	* -rw-rr
		root ±	wheel	t drwxr-xr-x
	dark-16.theme	root ±	wheel	‡ -rw-rr
	dark-256.theme	root ±	wheel	+ -rw-rr
	dark-blue-256.theme	root ±	wheel	‡ -rw-rw-r
	dark-green-256.theme	root ±	wheel	‡ -rw-rw-r
	dark-red-256.theme	root ±	wheel	* -rw-rw-r
	holidays-SE.rc	root 🗄	wheel	‡ -rw-rr
	M holidays-US.rc	root ±	wheel	* -rw-rr
	M Pilight-16.theme	root 🛔	wheel	‡ -rw-rr
	light-256.theme	root ±	wheel	* -rw-rr
	README	root 🙏	wheel	‡ -rw-rr
	Scripts	root ‡	wheel	t drwxr-xr-x
	M V bash	root 💠	wheel	t drwxr-xr-x
	task completion.sh	root ‡	wheel	‡ -rw-rr
		root ‡	wheel	drwxr-xr-x
	✓ The sector of the sector	root ‡	wheel	drwxr-xr-x
	✓ task.vim	root ‡	wheel	‡ -rw-rr
	README	root ‡	wheel	‡ -rw-rr
	✓ ▼ syntax	root ‡	wheel	drwxr-xr-x
	taskdata.vim	root 🕴	wheel	‡ -rw-rr
	✓ taskedit.vim	root ‡	wheel	‡ -rw-rr
	✓ taskrc.vim	root 🕴	wheel	‡ -rw-rr
	✓ ▼ 2sh	root ‡	wheel	# drwxr-xr-x
	✓ _task	root ‡	wheel	‡ -rw-rr
	v man	642 ‡	542	‡ drwxrwsr-x
	🗹 🔍 🔽 📩 man 1	642 ‡	542	# drwxr-sr-x
	☑ task.1	root ‡	542	‡ -rw-rr
	🗹 🔻 🔚 man 5	642 ‡	542	# drwxr-sr-x
	✓ task-color.5	root ‡	542	‡ -rw-rr
	☑ task-faq.5	root ‡	542	‡ -rw-rr
	🗹 📄 task-tutorial.5	root ‡	542	‡ -rw-rr
	☑ 🔮 taskrc.5	root 🕴	542	‡ -rw-rr
	Include root in package	File Filters	Apply	Recommendations
	Owner: Read Write Execute			
	Group: Read Write Execute		Owner:	No Selection 🕴
	Others O Read O Write O Execute		Group:	No Selection 🛟
	Others: Kead Write Execute			

- 6.12 Click on the "Components" tab/button, make sure it is all blank.
- 6.13 Click on the "Scripts" tab/button, make sure it is all blank.

7. Building the package

7.1 Click on the "Build" hammer icon to build the package.

Provide a filename of "task-1.9.1-sl.pkg" for Snow Leopard (10.6), or "task-1.9.2.pkg" for Leopard (10.5), and save it somewhere, for example the Desktop.

All should succeed. You can click on the "Return" button to end the build phase, and you can quit PackageManager, but make sure you save the changes you made, because you may need to go through the whole process again, if there is an emergency change, or the package is somehow corrupt.



8. Test the package

8.1 Double-click on the package you just created, and install taskwarrior. You should see the README file in the UI, and the COPYING file on another page. It should succeed.















8.2 Run the following commands to test the installation

\$ /usr/local/bin/task version

Taskwarrior 1.9.2 built for darwin-ncurses Copyright (C) 2006 - 2010 P. Beckingham, F. Hernandez.

Taskwarrior may be copied only under the terms of the GNU General Public License, which may be found in the taskwarrior source kit.

```
Documentation for taskwarrior can be found using 'man task',
'man taskrc', 'man task-tutorial', 'man task-color', 'man task-faq'
or at <a href="http://taskwarrior.org">http://taskwarrior.org</a>
```

\$ man task
...

... task 1.9.2 2010-05-22

task 1.9.2

The man page should list 1.9.2 as the version number, but the date will be different.

9. Email the package to Fredde.
