



Writing Better AutoPkg Recipes With

Recipe Robot



Elliot Jordan
Emeryville, CA



Eldon Ahrold
St. Louis, MO

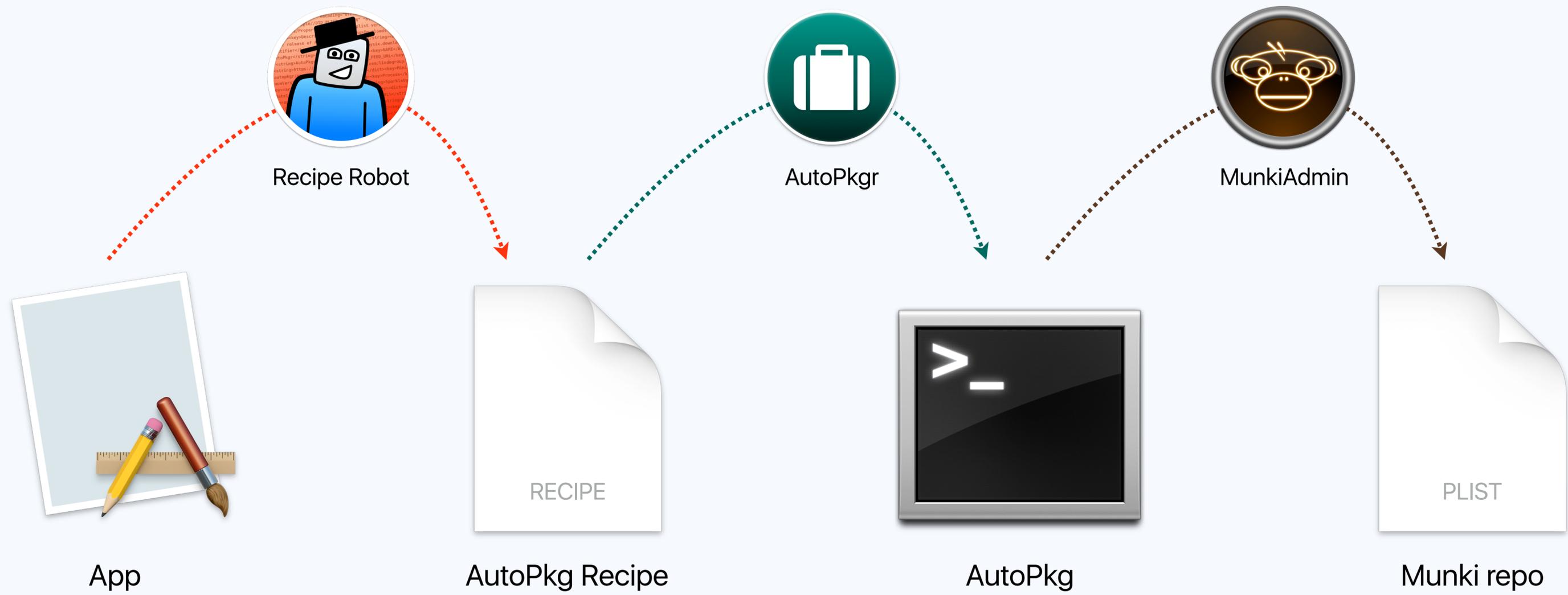


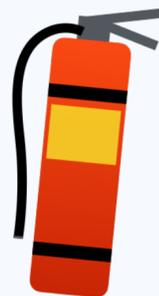
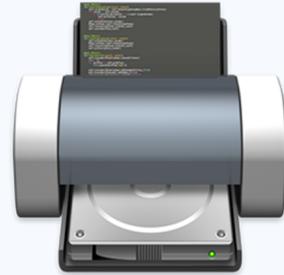
Shea Craig
Cary, NC

Agenda

- ❑ What is Recipe Robot?
- ❑ How does it work?
- ❑ First, some time-saving tips
- ❑ Learn to read/write recipes
- ❑ Audit existing AutoPkg recipes
- ❑ Augment recipes with URLTextSearcher
- ❑ Create child recipes of existing recipes (!)
- ❑ Publishing your recipes
- ❑ Troubleshooting
- ❑ Questions & Answers









Mac App

*Drag and drop
Friendly, easy, simple
Sweet icon*



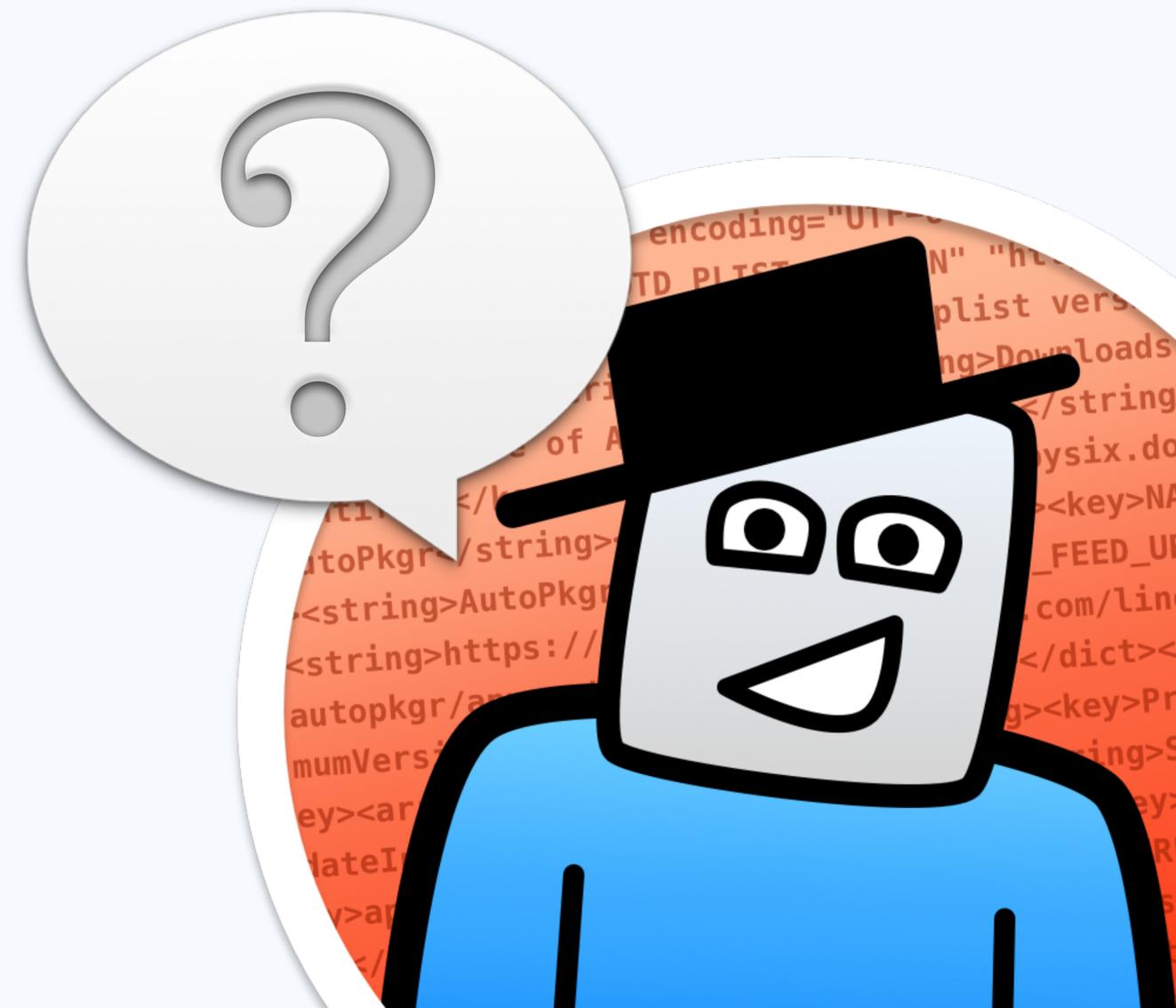
Python Script

*Stable, tested
Gets new features first
Easier troubleshooting
Geek cred 🧐*

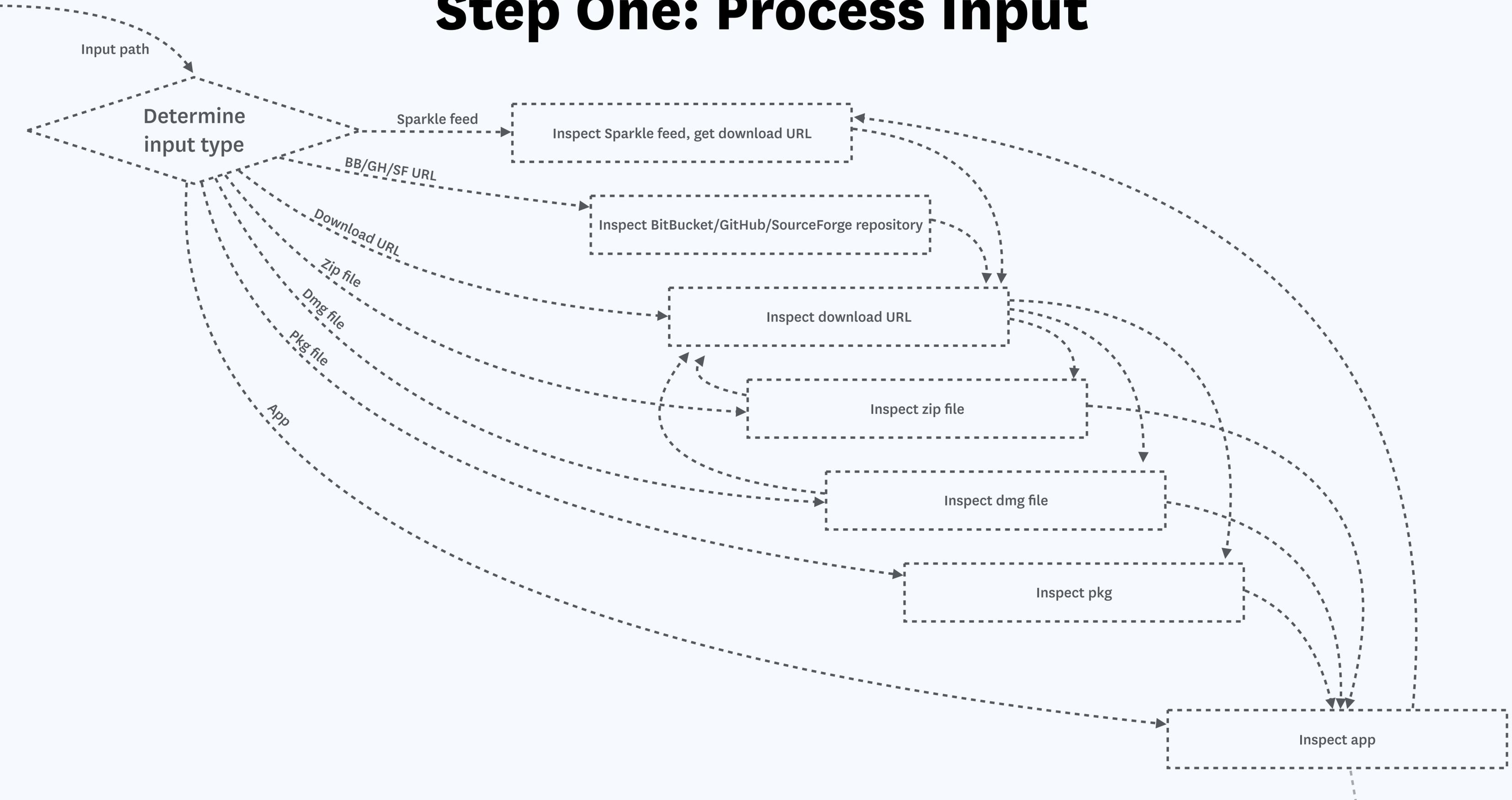
<http://j.mp/recipe-robot>

How does it work?

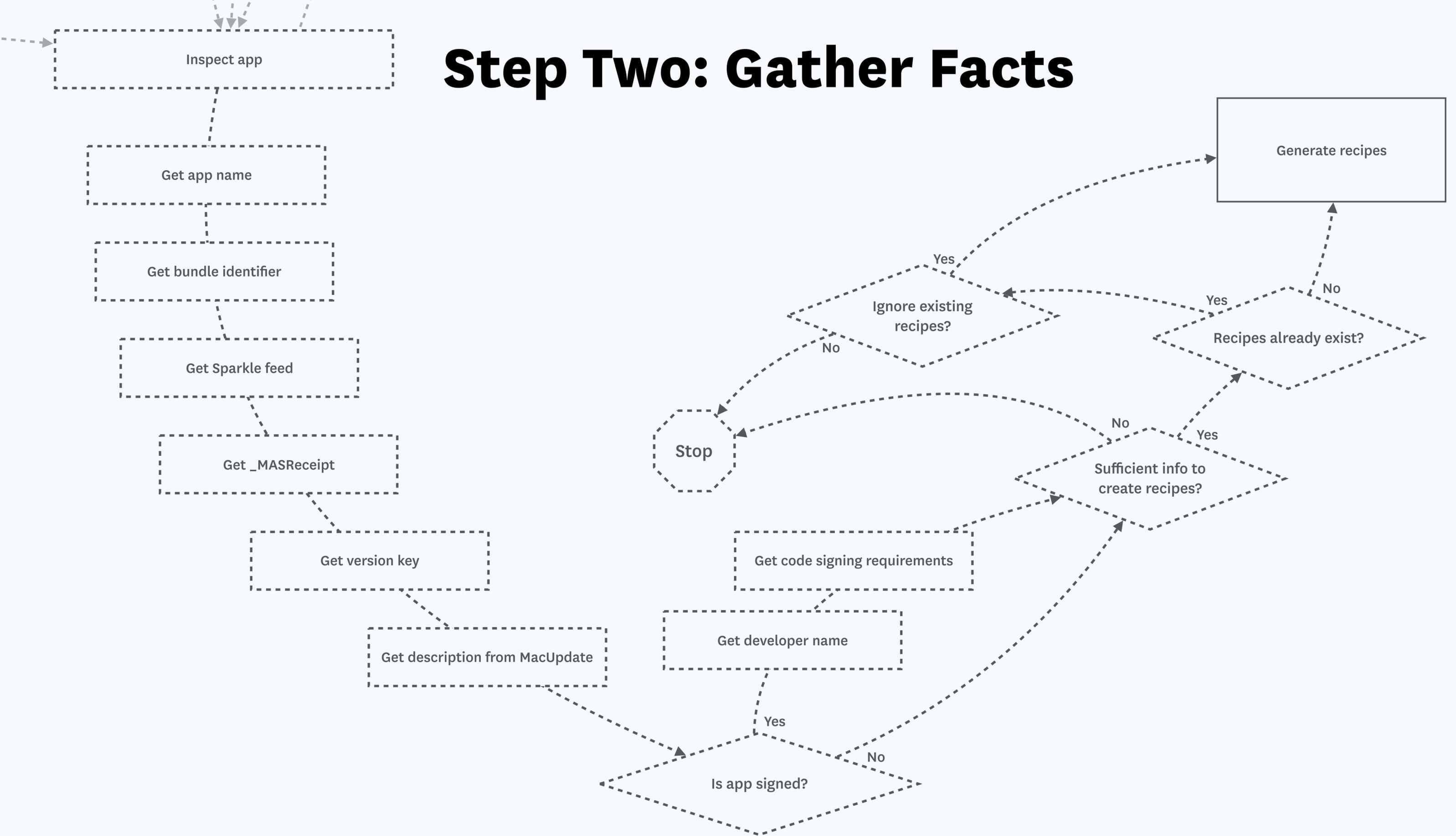
It's not magic. It's Python.



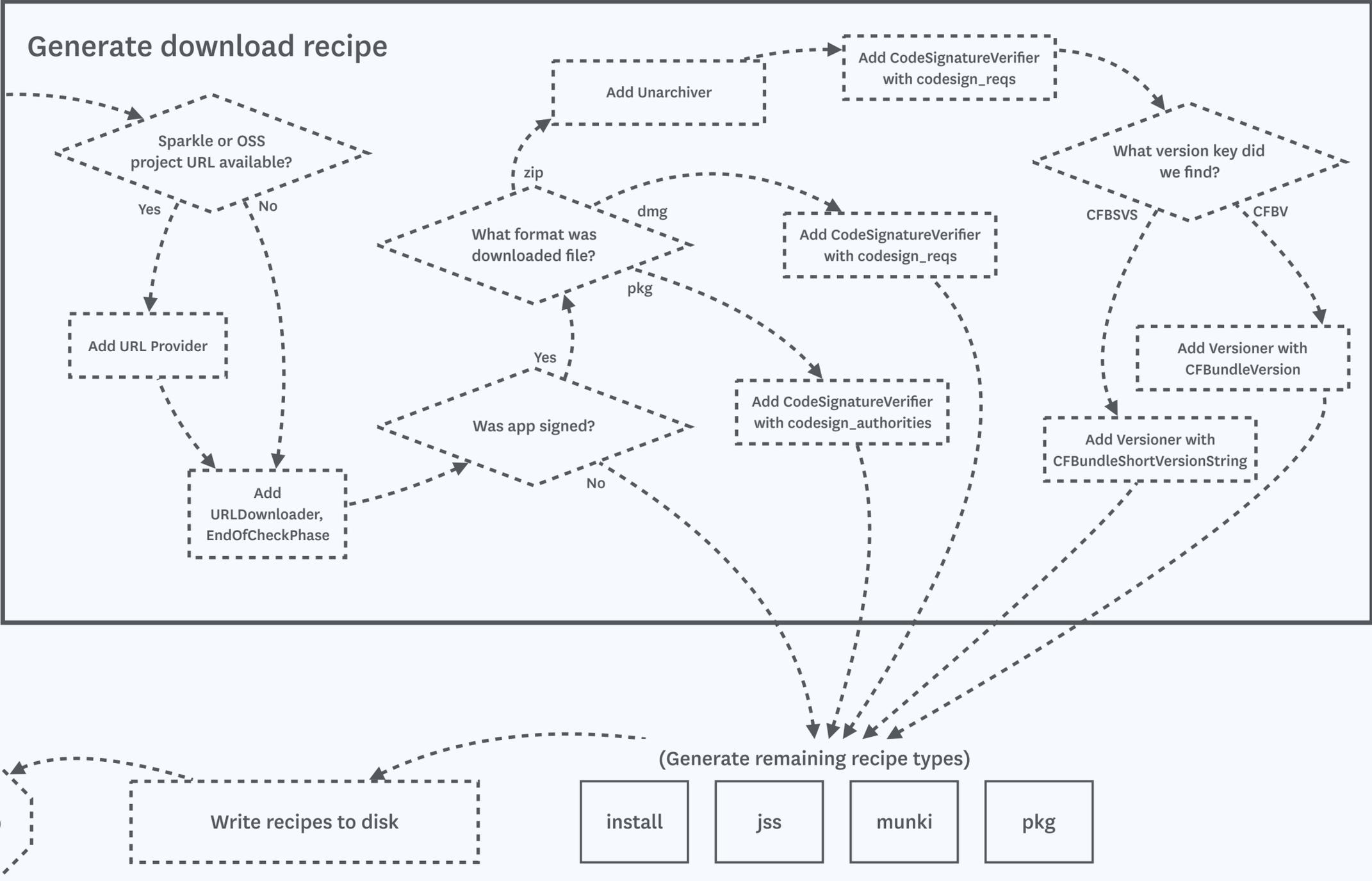
Step One: Process Input



Step Two: Gather Facts



Step Three: Generate Recipes



It's not magic.

Recipe Robot still needs you to review, test,
and customize what it creates.



`com.github.you.download.Foo`



`com.github.you.pkg.Foo`



`com.github.you.jss.Foo`

It's not magic.

Recipe Robot (still) only builds complete recipe sets.



...but we'll return to this later!

It's not magic.

Recipe Robot won't work for every app every time.



Before we go farther:

Some basic time-saving tips



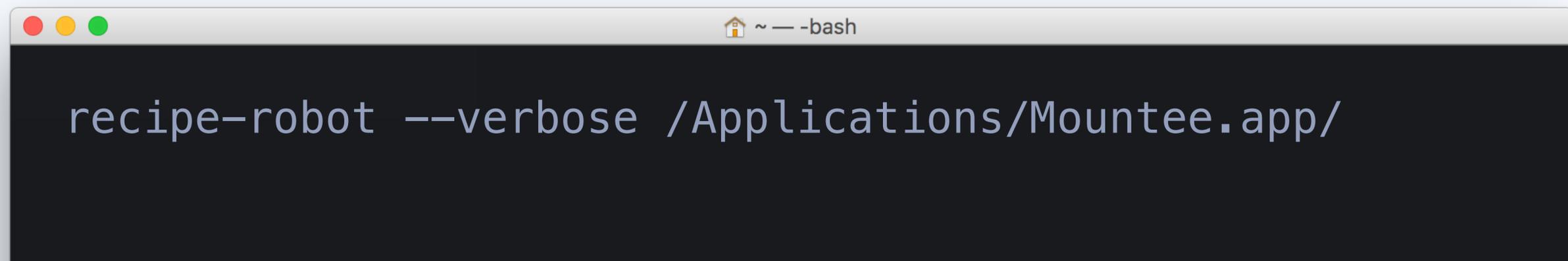
Customize your `bash_profile`

1. Create a file at `~/.bash_profile`, if it doesn't already exist.
2. Edit the file with a text editor. Add this line:

```
alias recipe-robot="/Applications/Recipe\ Robot.app/Contents/Resources/scripts/recipe-robot"
```

3. Save.

And now you can do this:

A screenshot of a macOS terminal window. The title bar shows a home icon, a tilde (~), and the text "-bash". The terminal content shows the command "recipe-robot --verbose /Applications/Mountee.app/" being entered. The window has standard macOS window controls (red, yellow, green buttons) in the top-left corner.

```
recipe-robot --verbose /Applications/Mountee.app/
```

Use a good text editor



TextWrangler

barebones.com



Atom

atom.io



TextMate

macromates.com



BBEdit

barebones.com



Sublime Text

sublimetext.com

...and literally dozens more.

Use a good text editor

Syntax highlighting makes recipes more readable...

The image shows a text editor window with a sidebar on the left displaying a file tree. The main editor area shows a recipe file named `BitTorrentSync.download.recipe` with the following content:

```
21 >> <array>
22 >> <dict>
23 >> >> <key>Arguments</key>
24 >> >> <dict>
25 >> >> >> <key>filename</key>
26 >> >> >> <string>BitTorrent-Sync.dmg</string>
27 >> >> >> <key>url</key>
28 >> >> >> <string>%DOWNLOAD_URL%</string>
29 >> >> </dict>
30 >> >> <key>Processor</key>
31 >> >> <string>URLDownloader</string>
32 >> </dict>
33 >> <dict>
34 >> >> <key>Processor</key>
35 >> >> <string>EndOfCheckPhase</string>
36 >> </dict>
37 >> <dict>
38 >> >> <key>Arguments</key>
39 >> >> <dict>
40 >> >> >> <key>input_path</key>
41 >> >> >> <string>%pathname%/BitTorrent Sync.app</string>
42 >> >> >> <key>requirement</key>
43 >> >> >> <string>identifier "com.bittorrent.Sync" and anchor apple generic
44 >> >> >> and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and
45 >> >> >> certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
46 >> >> >> certificate leaf[subject.OU] = "2953Z5SZSK"</string>
```

Annotations with arrows pointing to the code:

- keys**: Points to the `<key>` tags in the first dictionary.
- tags**: Points to the `<string>` tags in the first dictionary.
- values**: Points to the string values in the second dictionary.
- invisibles**: Points to the vertical line of indentation characters (spaces and tabs) used for structure.

Use a good text editor

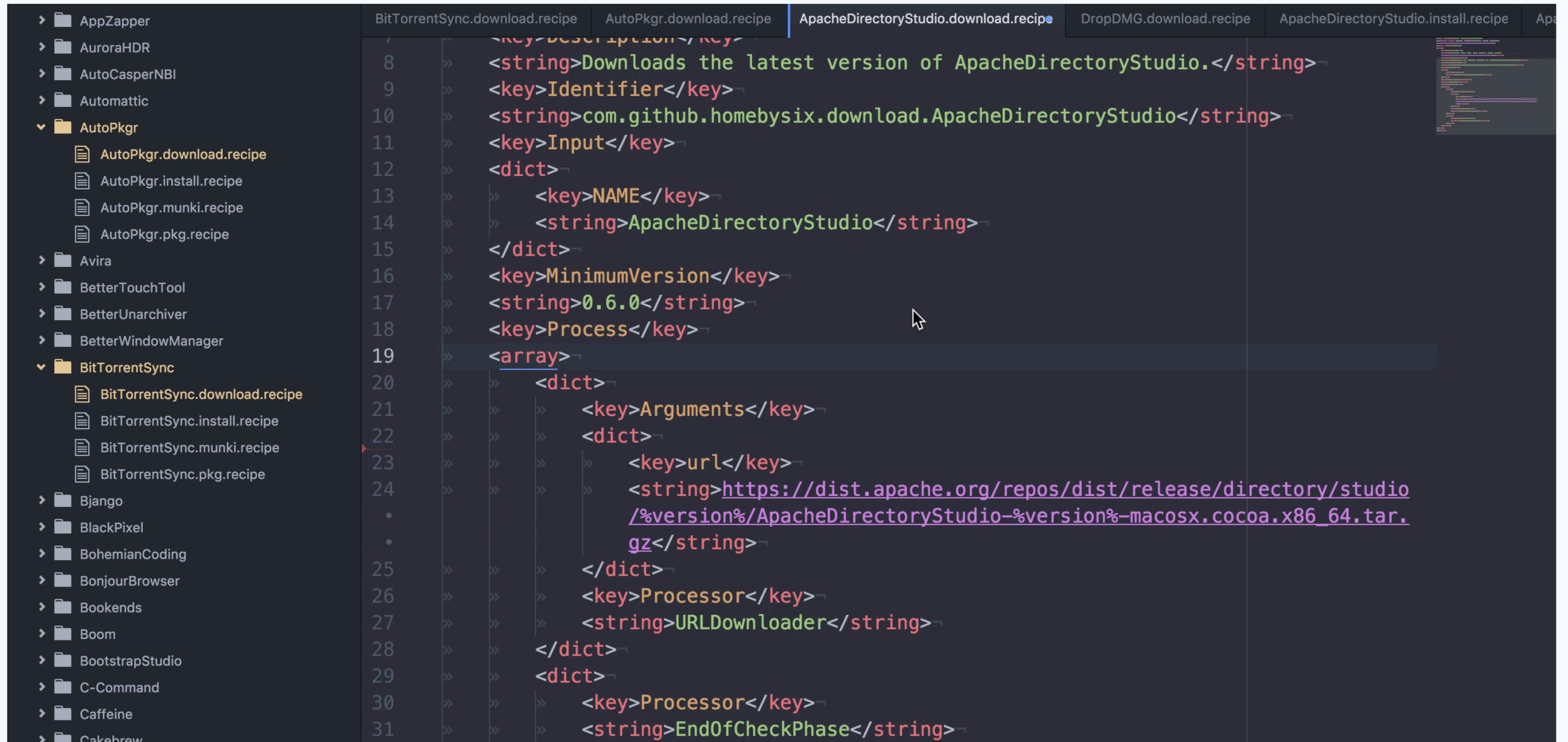
...and makes some basic mistakes obvious.

The screenshot shows a text editor window titled "AutoPkgr.download.recipe" with a file explorer on the left. The code is XML and contains several errors. A callout box labeled "bad tag" points to a closing tag that is not properly closed.

```
19 <string>0.0.1</string>
20 <key>Process</key>
21 <array>
22 >> <dict>
23 >>> <key>Arguments</key>
24 >>> <dict>
25 >>>> <key>appcast_url</key>
26 >>>> <string>%SPARKLE_FEED_URL%</string>
27 >>> </dict>
28 >>> <key>Processor</key>
29 >>> <string>SparkleUpdateInfoProvider</string>
30 >> </dict>
31 >> <dict>
32 >>> <key>Arguments</key>
33 >>> <dict>
34 >>>> <key>filename</key>
35 >>>> <string>%NAME%-%version%.dmg</string>
36 >>> </dict>
37 >>> <key>Processor</key>
38 >>> <string>URLDownloader</string>
39 >> </dict>
40 >> <dict>
41 >>> <key>Processor</key>
42 >>> <string>EndOfCheckPhase</string>
43 >> </dict>
44 >> <dict>
45 >>> <key>Arguments</key>
```

Use a good text editor

Property list auto-completion can do some work for you.



The screenshot shows a text editor with a dark theme. On the left is a sidebar with a file tree. The main editor area displays XML code for a recipe. The code is as follows:

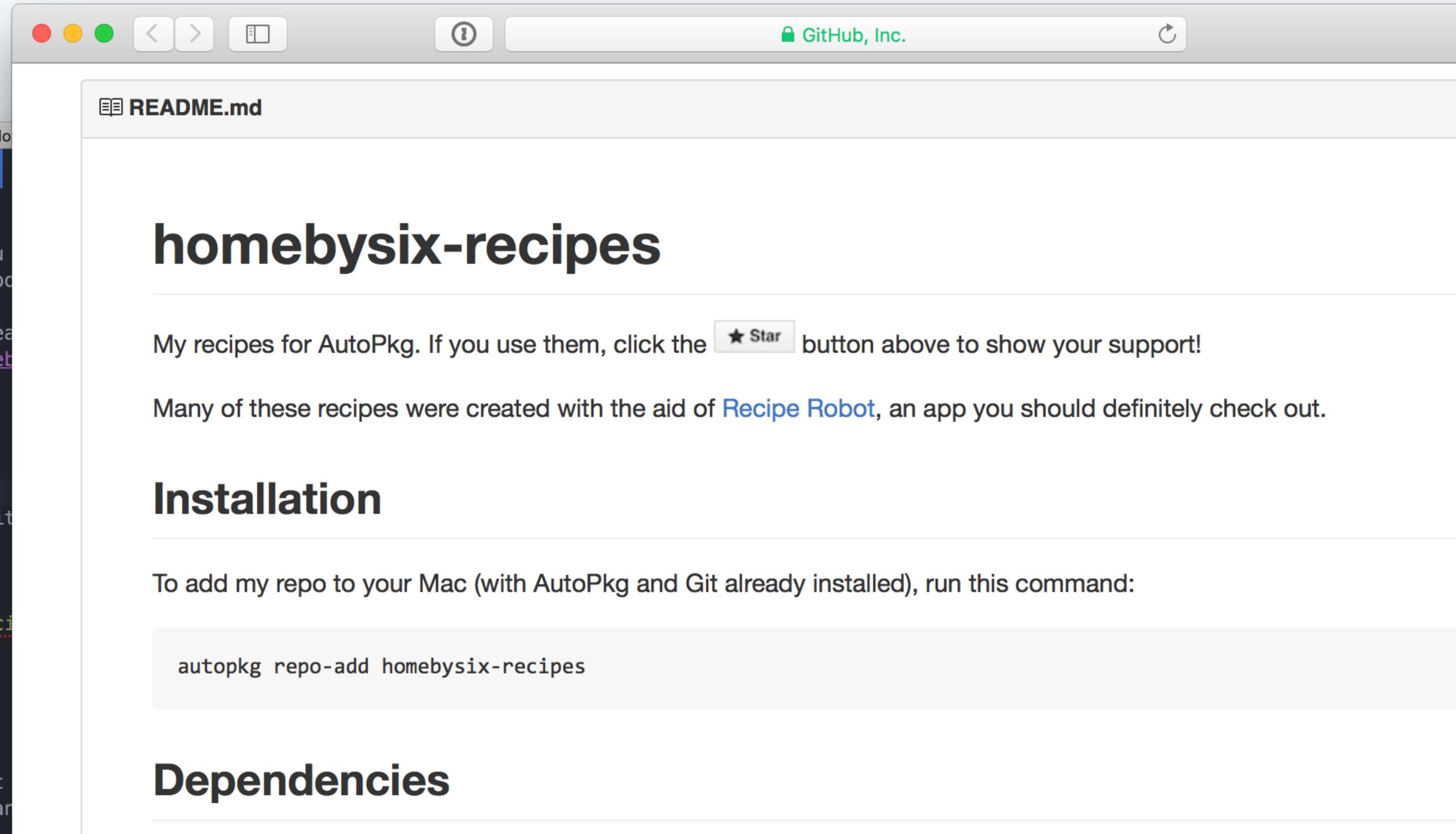
```
7 <key>DESCRIPTION</key>
8 <string>Downloads the latest version of ApacheDirectoryStudio.</string>
9 <key>Identifier</key>
10 <string>com.github.homebysix.download.ApacheDirectoryStudio</string>
11 <key>Input</key>
12 <dict>
13   <key>NAME</key>
14   <string>ApacheDirectoryStudio</string>
15 </dict>
16 <key>MinimumVersion</key>
17 <string>0.6.0</string>
18 <key>Process</key>
19 <array>
20   <dict>
21     <key>Arguments</key>
22     <dict>
23       <key>url</key>
24       <string>https://dist.apache.org/repos/dist/release/directory/studio/%version%/ApacheDirectoryStudio-%version%-macosx.cocoa.x86\_64.tar.gz</string>
25     </dict>
26     <key>Processor</key>
27     <string>URLDownloader</string>
28   </dict>
29 <dict>
30   <key>Processor</key>
31   <string>EndOfCheckPhase</string>
```

The code is color-coded: keys are in orange, strings in green, and arrays/dicts in blue. A mouse cursor is positioned over the code on line 18. The sidebar on the left shows a tree view with folders like 'AppZapper', 'AuroraHDR', 'AutoCasperNBI', 'Automattic', 'AutoPkgr', 'Avira', 'BetterTouchTool', 'BetterUnarchiver', 'BetterWindowManager', 'BitTorrentSync', 'Bjango', 'BlackPixel', 'BohemianCoding', 'BonjourBrowser', 'Bookends', 'Boom', 'BootstrapStudio', 'C-Command', 'Caffeine', and 'Cakebrew'. The 'AutoPkgr' folder is expanded, showing files like 'AutoPkgr.download.recipe', 'AutoPkgr.install.recipe', 'AutoPkgr.munki.recipe', and 'AutoPkgr.pkg.recipe'. The 'BitTorrentSync' folder is also expanded, showing files like 'BitTorrentSync.download.recipe', 'BitTorrentSync.install.recipe', 'BitTorrentSync.munki.recipe', and 'BitTorrentSync.pkg.recipe'.

Master Markdown

Your README.md will be a page-turner.

```
README.md — /Users/elliott/Develo
BitTorrentSync.download.recipe  AutoPkgr.download.recipe
1 # homebysix-recipes
2 -
3 My recipes for AutoPkg. If you
4 • button above to show your supp
5 -
6 Many of these recipes were crea
7 • Robot](https://github.com/homeb
8 • check out.
9 -
10 ## Installation
11 -
12 To add my repo to your Mac (wit
13 • command:
14 -
15 ```
16 autopkg repo-add homebysix-recipe
17 ```
18 -
19 ## Dependencies
20 -
21 Some of my recipes have parent
22 • in one shot, here is the comman
23 -
```

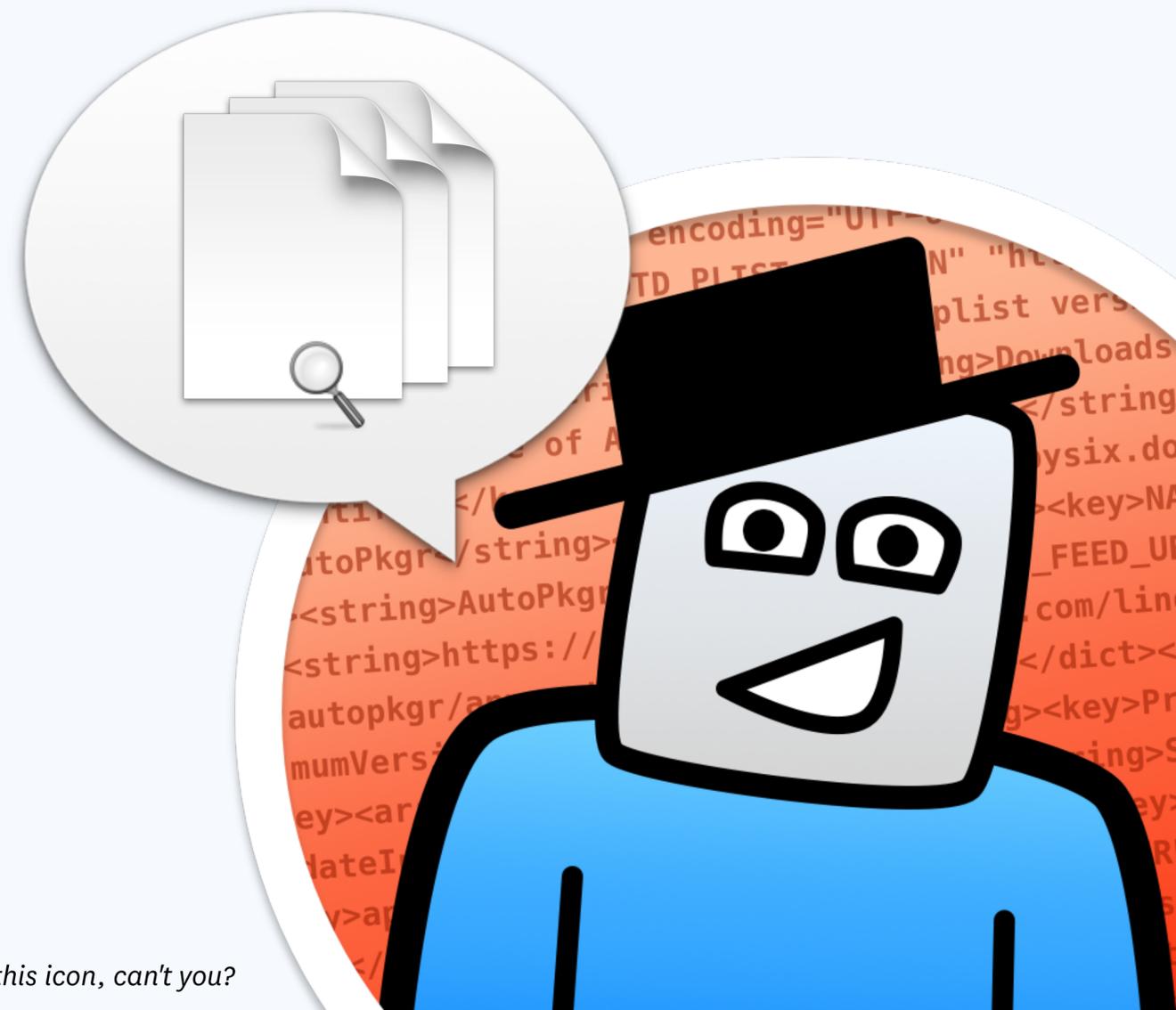


Onward!

- What is Recipe Robot?
- How does it work?
- First, some time-saving tips
- Learn to read/write recipes
- Audit existing AutoPkg recipes
- Augment recipes with URLTextSearcher
- Create child recipes of existing recipes (!)
- Publishing your recipes
- Troubleshooting
- Questions & Answers

Using Recipe Robot to

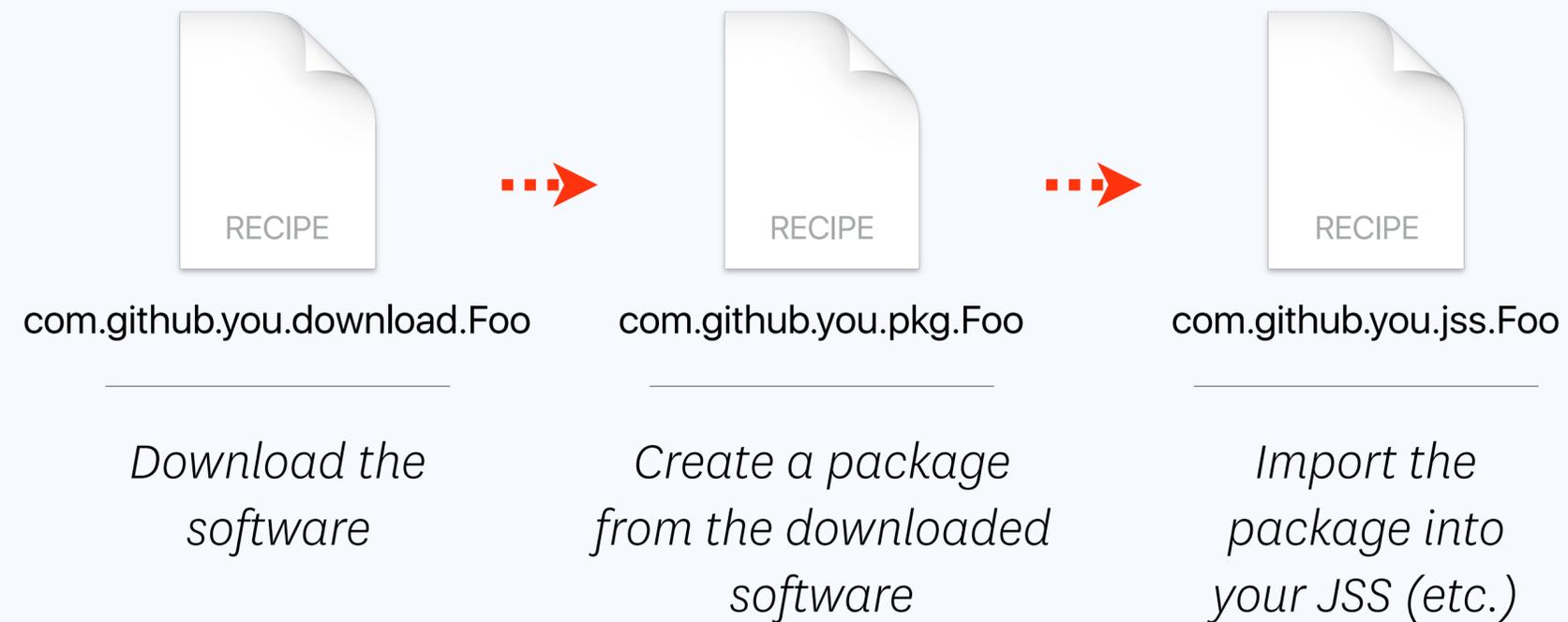
Learn to read/write AutoPkg recipes



You can tell I like this icon, can't you?

Separation of Responsibility

Each recipe accomplishes a discrete goal.



Separation of Responsibility

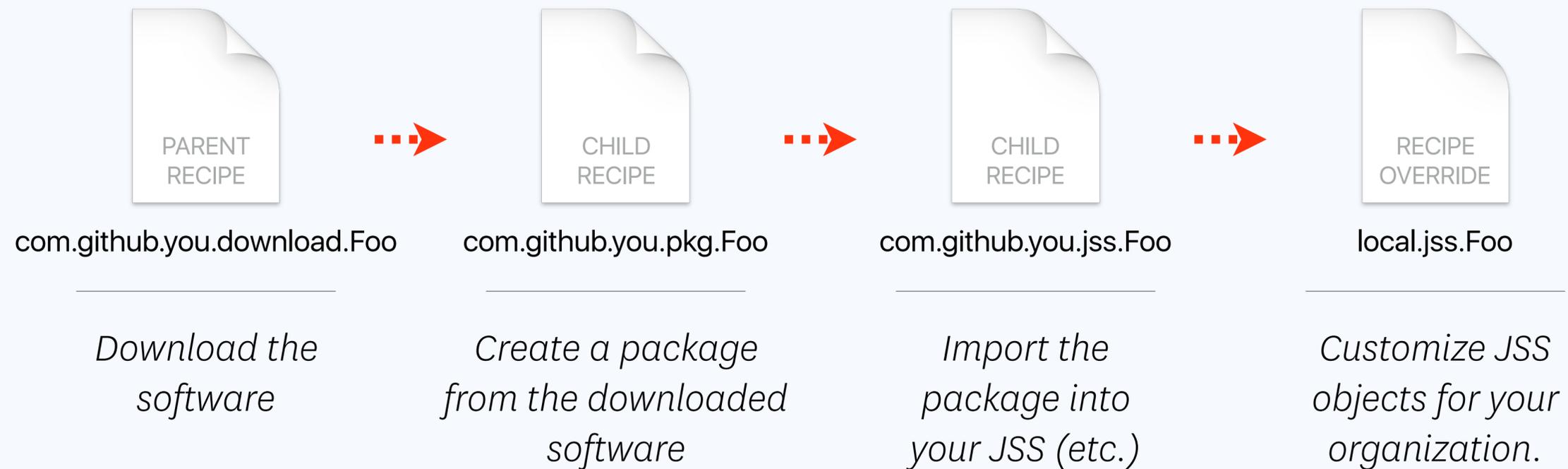
Recipe Robot respects this separation.

You'll find the goal of each recipe right there in the Description.

```
4 <dict>
5   <key>Comment</key>
6   <string>Created with Recipe Robot v1.0.2 (https://github.com/homebysix/recipe-robot)</
string>
7   <key>Description</key>
8   <string>Downloads the latest version of Wondershare Filmora.</string>
9   <key>Identifier</key>
10  <string>com.github.homebysix.download.Filmora</string>
11  <key>Input</key>
12  <dict>
13    <key>NAME</key>
14    <string>Wondershare Filmora</string>
15  </dict>
16  <key>MinimumVersion</key>
```

Parent/Child Relationship

Recipes build on other recipes.



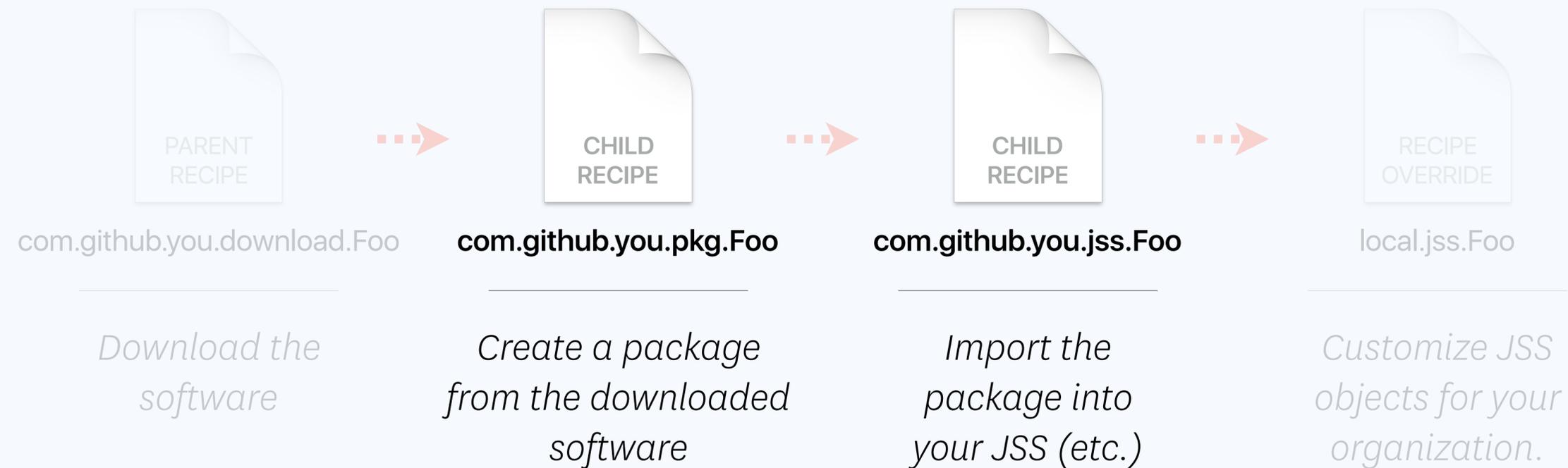
Parent/Child Relationship

Parent recipes don't depend upon any other recipes.



Parent/Child Relationship

Child recipes perform additional processes on output from other recipes.



Parent/Child Relationship

Recipe overrides ARE child recipes!

(But instead of additional processes, they just change input variables.)



Parent/Child Relationship

The ParentRecipe key leads you back to the start.

```
30     <string>%NAME%</string>
31     <key>unattended_install</key>
32     <true/>
33 </dict>
34 </dict>
35 <key>MinimumVersion</key>
36 <string>0.6.1</string>
37 <key>ParentRecipe</key>
38 <string>com.github.homebysix.download.Filmora</string>
39 <key>Process</key>
40 <array>
41   <dict>
42     <key>Arguments</key>
43     <dict>
```

How do we know?

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Comment</key>
6   <string>Generated by Recipe Robot v0.0.3 (https://github.com/homebysix/recipe-robot)</string>
7   <key>Description</key>
8   <string>Downloads the latest version of TMNotifier.</string>
9   <key>Identifier</key>
10  <string>com.github.homebysix.download.TMNotifier</string>
11  <key>Input</key>
12  <dict>
13    <key>NAME</key>
14    <string>TMNotifier</string>
15    <key>SPARKLE_FEED_URL</key>
16    <string>https://tmnotifier.com/appcast.xml</string>
17  </dict>
18  <key>MinimumVersion</key>
19  <string>0.6.1</string>
20  <key>Process</key>
21  <array>
22    <dict>
23      <key>Arguments</key>
24      <dict>
25        <key>appcast_url</key>
26        <string>%SPARKLE_FEED_URL%</string>
27      </dict>
28      <key>Processor</key>
29      <string>SparkleUpdateInfoProvider</string>
30    </dict>
31    <dict>
32      <key>Arguments</key>
33      <dict>
34        <key>filename</key>
35        <string>%NAME%-%version%.dmg</string>
36      </dict>
37      <key>Processor</key>
38      <string>URLDownloader</string>
39    </dict>
40    <dict>
41      <key>Processor</key>
42      <string>EndOfCheckPhase</string>
43    </dict>
44    <dict>
45      <key>Arguments</key>
46      <dict>
47        <key>input_path</key>
48        <string>%pathname%/%NAME%.app</string>
49        <key>requirement</key>
50        <string>anchor apple generic and identifier "com.everythingisgray.TMNotifier" and (certificate leaf[field.1.2.840.113635.100.6.1.9] /* exists */
51      </dict>
52      <key>Processor</key>
53      <string>CodeSignatureVerifier</string>
54    </dict>
55  </array>
56 </dict>
57 </plist>
```

How do we know?

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Comment</key>
6   <string>Generated by Recipe Robot v0.0.3 (https://github.com/homebysix/recipe-robot)</string>
7   <key>Description</key>
8   <string>Downloads the latest version of TMNotifier.</string>
9   <key>Identifier</key>
10  <string>com.github.homebysix.download.TMNotifier</string>
11  <key>Input</key>
12  <dict>
13    <key>NAME</key>
14    <string>TMNotifier</string>
15    <key>SPARKLE_FEED_URL</key>
16    <string>https://tmnotifier.com/appcast.xml</string>
17  </dict>
18  <key>MinimumVersion</key>
19  <string>0.6.1</string>
20  <key>Process</key>
21  <array>
22    <dict>
23      <key>Arguments</key>
24      <dict>
25        <key>appcast_url</key>
26        <string>%SPARKLE_FEED_URL%</string>
27      </dict>
28      <key>Processor</key>
29      <string>SparkleUpdateInfoProvider</string>
30    </dict>
31    <dict>
32      <key>Arguments</key>
33      <dict>
34        <key>filename</key>
35        <string>%NAME%-%version%.dmg</string>
36      </dict>
37      <key>Processor</key>
38      <string>URLDownloader</string>
39    </dict>
40    <dict>
41      <key>Processor</key>
42      <string>EndOfCheckPhase</string>
43    </dict>
44    <dict>
45      <key>Arguments</key>
46      <dict>
47        <key>input_path</key>
48        <string>%pathname%/%NAME%.app</string>
49        <key>requirement</key>
50        <string>anchor apple generic and identifier "com.everythingisgray.TMNotifier" and (certificate leaf[field.1.2.840.113635.100.6.1.9] /* exists */
51      </dict>
52      <key>Processor</key>
53      <string>CodeSignatureVerifier</string>
54    </dict>
55  </array>
56 </dict>
57 </plist>
```

<key>Processor</key>
<string>SparkleUpdateInfoProvider</string>

<key>Processor</key>
<string>URLDownloader</string>

<key>Processor</key>
<string>EndOfCheckPhase</string>

<key>Processor</key>
<string>CodeSignatureVerifier</string>

Processor Groupings

Processors are the tasks that combine to accomplish the goal.



Processor Groupings

Many processors appear in the same sequence over and over.



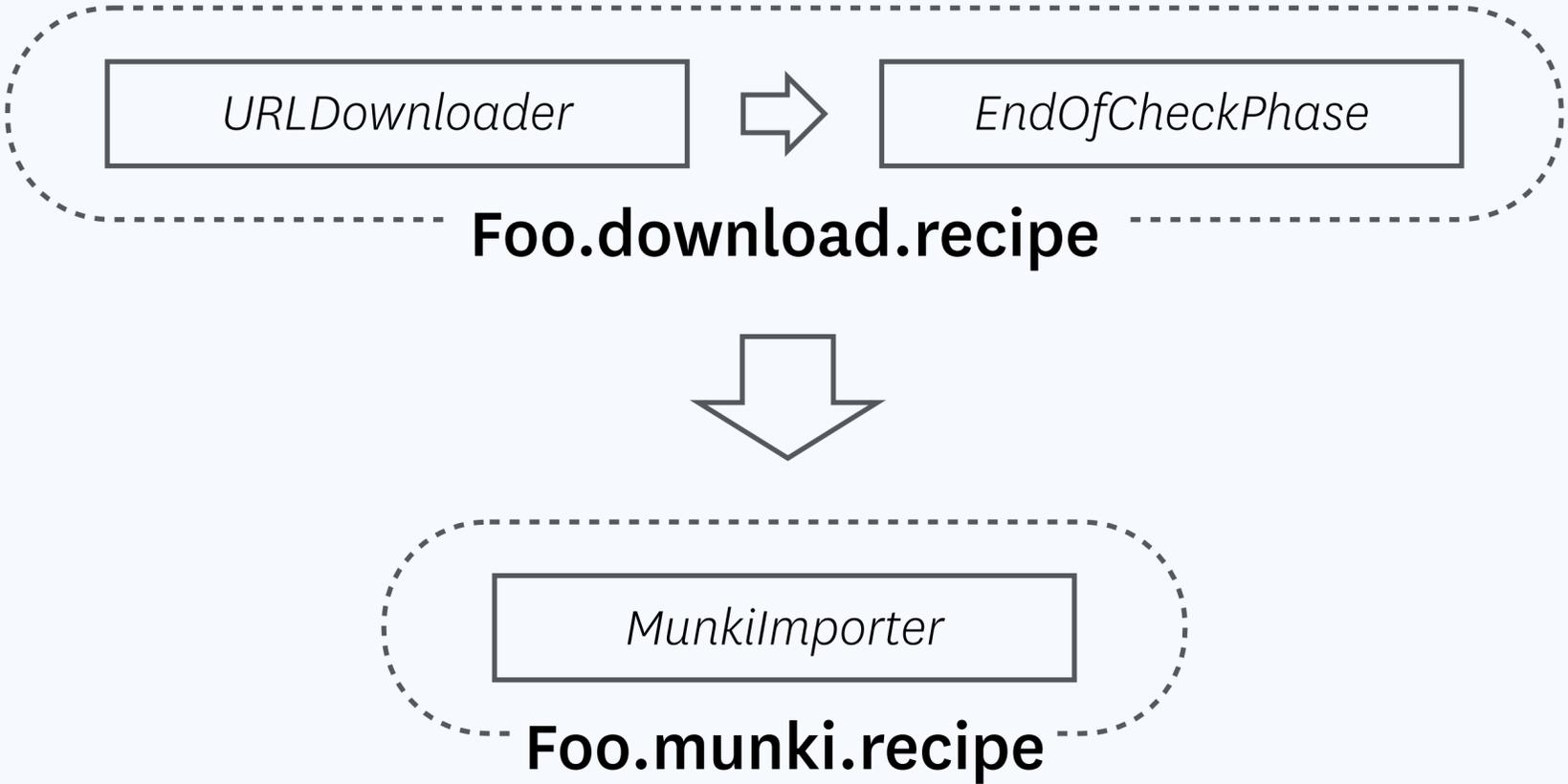
Processor Groupings

Recipe Robot knows this and places processors accordingly.

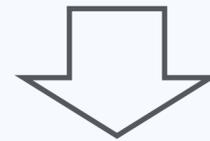
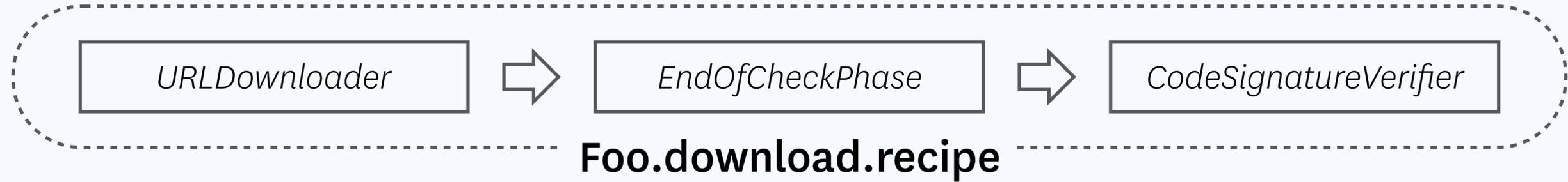


Processor Groupings

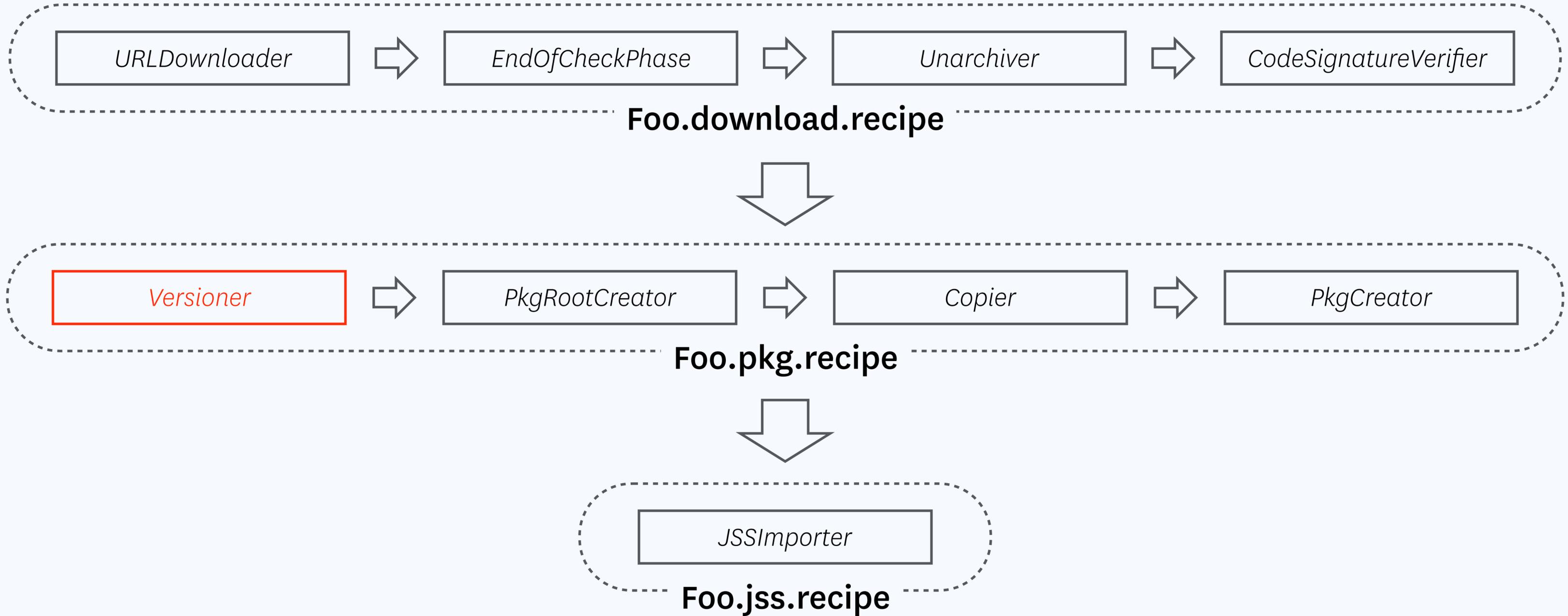
A chain of processors is created as child recipes build on parent recipes.



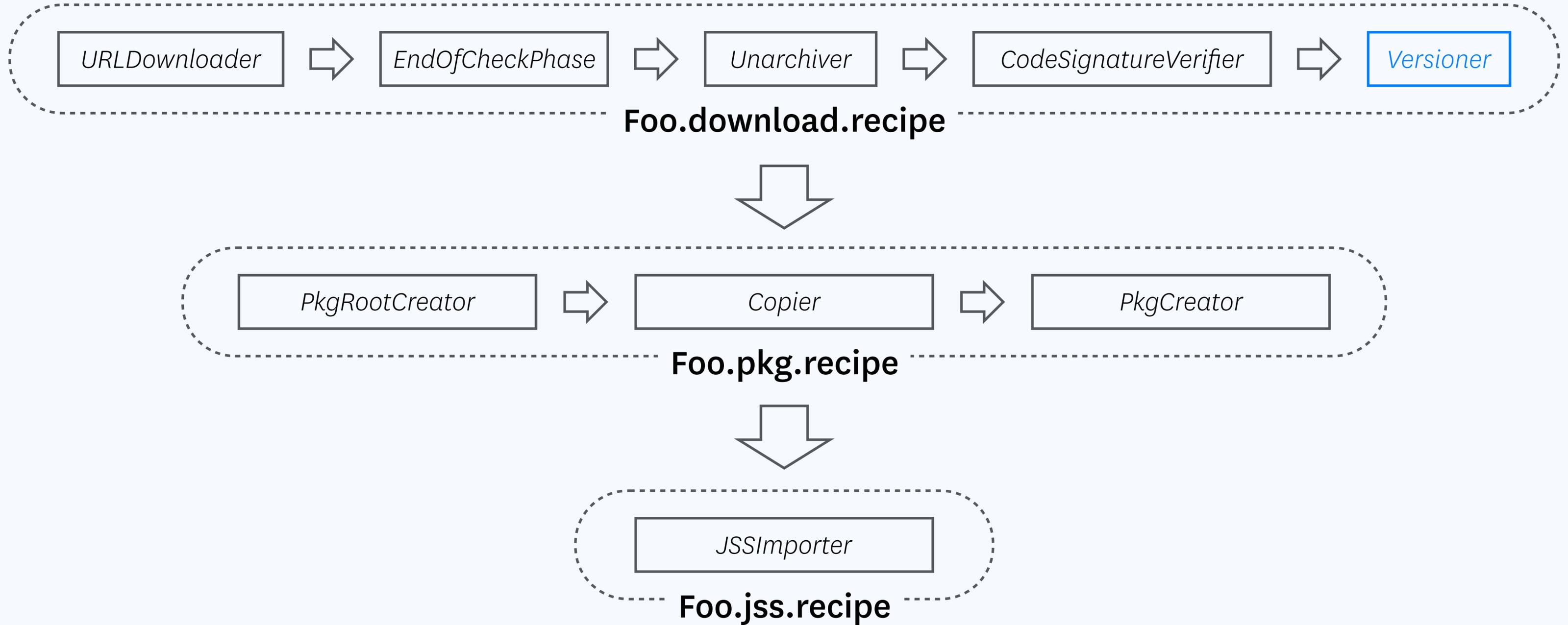
Processor Groupings



Processor Groupings



Processor Groupings



Performing a "diff" on two recipes

The screenshot shows a side-by-side comparison of two recipe files in Kaleidoscope. The left pane shows the 'Tower' recipe, and the right pane shows the 'uBar' recipe. The diff highlights several differences:

- Line 6: The 'Created with Recipe Robot' version is v0.0.4 for Tower and v0.0.6 for uBar.
- Line 8: The description for Tower is 'Downloads the latest version of Tower.', while for uBar it is 'Downloads the latest version of uBar.'.
- Line 10: The identifier for Tower is 'com.github.homebysix.download.Tower' and for uBar it is 'com.github.homebysix.download.uBar'.
- Line 13: Tower has a 'DOWNLOAD_URL' key with the value 'http://www.git-tower.com/download', while uBar has a 'NAME' key with the value 'uBar'.
- Line 15: Tower has a 'NAME' key with the value 'Tower', while uBar has a 'SPARKLE_FEED_URL' key with the value 'https://browsersoftware.com/appcasts/feeds/ubar/ubar3.xml'.
- Line 22: Tower has an 'Arguments' key with a dictionary containing 'filename', 'url', and 'DOWNLOAD_URL'. uBar has an 'Arguments' key with a dictionary containing 'appcast_url', 'SPARKLE_FEED_URL', and 'Processor'.
- Line 26: Tower's processor is 'URLDownloader', while uBar's processor is 'SparkleUpdateInfoProvider'.
- Line 27: Tower's filename is '%NAME%.zip', while uBar's filename is '%NAME%-%version%.zip'.

On the left: a direct download.

Uses DOWNLOAD_URL with URLDownloader.

On the right: a Sparkle download.

Uses SPARKLE_FEED_URL with SparkleUpdateInfoProvider.

Performing a "diff" on two recipes

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Comment</key>
6   * <string>Created with Recipe Robot v0.0.4 (https://github.com/homebysix/recipe-robot)</string>
7   <key>Description</key>
8   * <string>Downloads the latest version of Freeway Pro and creates a package.</string>
9   <key>Identifier</key>
10  * <string>com.github.homebysix.pkg.FreewayPro</string>
11  <key>Input</key>
12  <dict>
13    <key>BUNDLE_ID</key>
14    * <string>com.softpress.FreewayPro</string>
15    <key>NAME</key>
16    * <string>Freeway Pro</string>
17  </dict>
18  <key>MinimumVersion</key>
19  <string>0.5.0</string>
20  <key>ParentRecipe</key>
21  * <string>com.github.homebysix.download.FreewayPro</string>
22  <key>Process</key>
23  <array>
24    <dict>
25      <key>Arguments</key>
26      <dict>
27        <key>pkg_request</key>
28        <dict>
29          <key>chown</key>
30          <array>
31            <dict>
32              <key>group</key>
33              <string>admin</string>
34              <key>path</key>
35              <string>Applications</string>
36              <key>user</key>
37              <string>root</string>
38            </dict>
39          </array>
40          <key>id</key>
41          <string>%BUNDLE_ID%</string>
42          <key>options</key>
43          <string>purge_ds_store</string>
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Comment</key>
6   * <string>Generated by Recipe Robot v0.0.3 (https://github.com/homebysix/recipe-robot)</string>
7   <key>Description</key>
8   * <string>Downloads the latest version of Keyboard Maestro and creates a package.</string>
9   <key>Identifier</key>
10  * <string>com.github.homebysix.pkg.KeyboardMaestro</string>
11  <key>Input</key>
12  <dict>
13    <key>BUNDLE_ID</key>
14    * <string>com.stairways.keyboardmaestro.editor</string>
15    <key>NAME</key>
16    * <string>Keyboard Maestro</string>
17  </dict>
18  <key>MinimumVersion</key>
19  <string>0.5.0</string>
20  <key>ParentRecipe</key>
21  * <string>com.github.homebysix.download.KeyboardMaestro</string>
22  <key>Process</key>
23  <array>
24    <dict>
25      <key>Arguments</key>
26      <dict>
27        <key>pkg_request</key>
28        <dict>
29          <key>chown</key>
30          <array>
31            <dict>
32              <key>group</key>
33              <string>admin</string>
34              <key>path</key>
35              <string>Applications</string>
36              <key>user</key>
37              <string>root</string>
38            </dict>
39          </array>
40          <key>id</key>
41          <string>%BUNDLE_ID%</string>
42          <key>options</key>
43          <string>purge_ds_store</string>
```

On the left: Freeway Pro.

On the right: Keyboard Maestro.

No major differences at all!

Using Recipe Robot to

Audit existing AutoPkg recipes



Audit your existing AutoPkg recipes

1. Standardize the existing recipes with:

```
plutil -convert xml1 *.recipe
```

- Finds syntax errors.
- Converts leading spaces to leading tabs.
- Alphabetizes dictionary keys (*Arguments* before *Processor*).
- Removes `<!-- -->` comments (but keeps `<key>Comment</key>`).
- Standardizes *DOCTYPE* string.

Why do we do this?

- It will become **very** easy to compare Recipe Robot's recipes against your own.

Audit your existing AutoPkg recipes

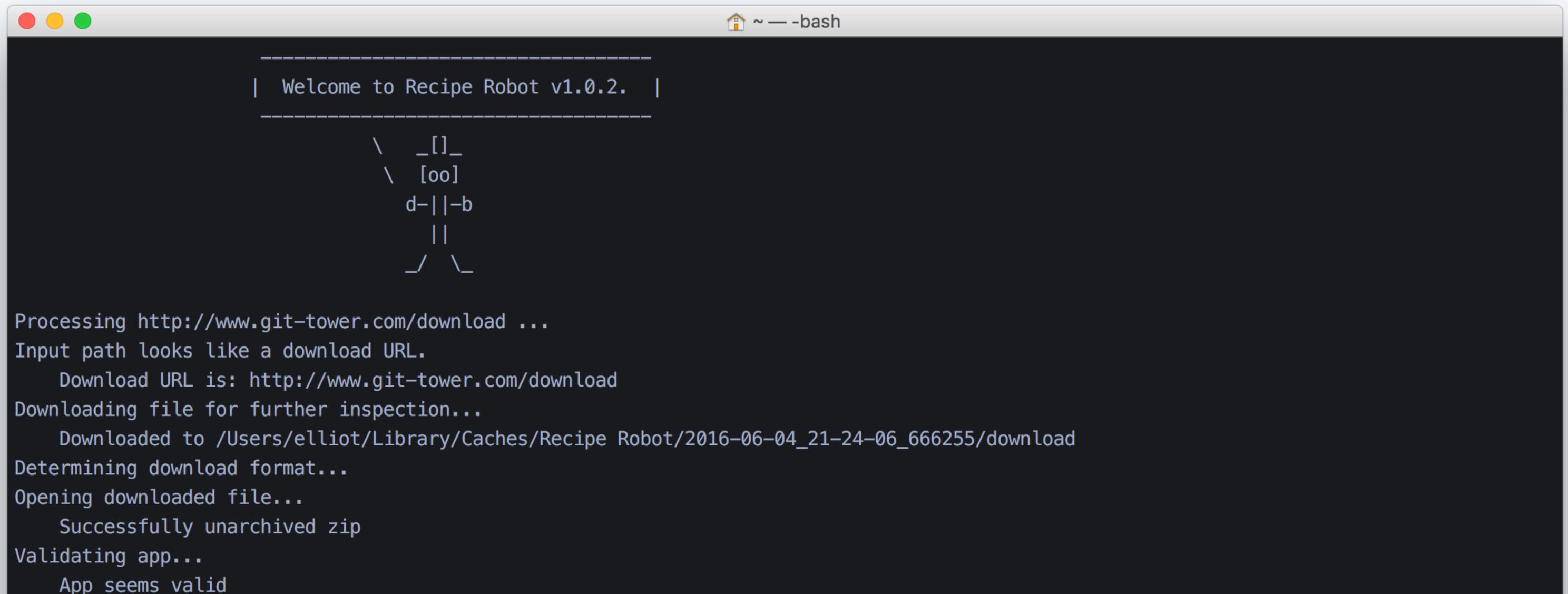
2. Locate and copy the download URL or Sparkle feed.

```
Tower.download.recipe
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Description</key>
6   <string>Downloads the latest version of Tower.</string>
7   <key>Identifier</key>
8   <string>com.github.homebysix.download.Tower</string>
9   <key>Input</key>
10  <dict>
11    <key>NAME</key>
12    <string>Tower</string>
13  </dict>
14  <key>MinimumVersion</key>
15  <string>0.3.0</string>
16  <key>Process</key>
17  <array>
18    <dict>
19      <key>Arguments</key>
20      <dict>
21        <key>filename</key>
22        <string>download.zip</string>
23        <key>url</key>
24        <string>http://www.git-tower.com/download</string>
25      </dict>
26      <key>Processor</key>
27      <string>URLDownloader</string>
28    </dict>
29  </array>
30 </dict>
```

Audit your existing AutoPkg recipes

3. Run that URL through Recipe Robot.

If your existing recipes are in the AutoPkg org on GitHub, use `--ignore-existing`.

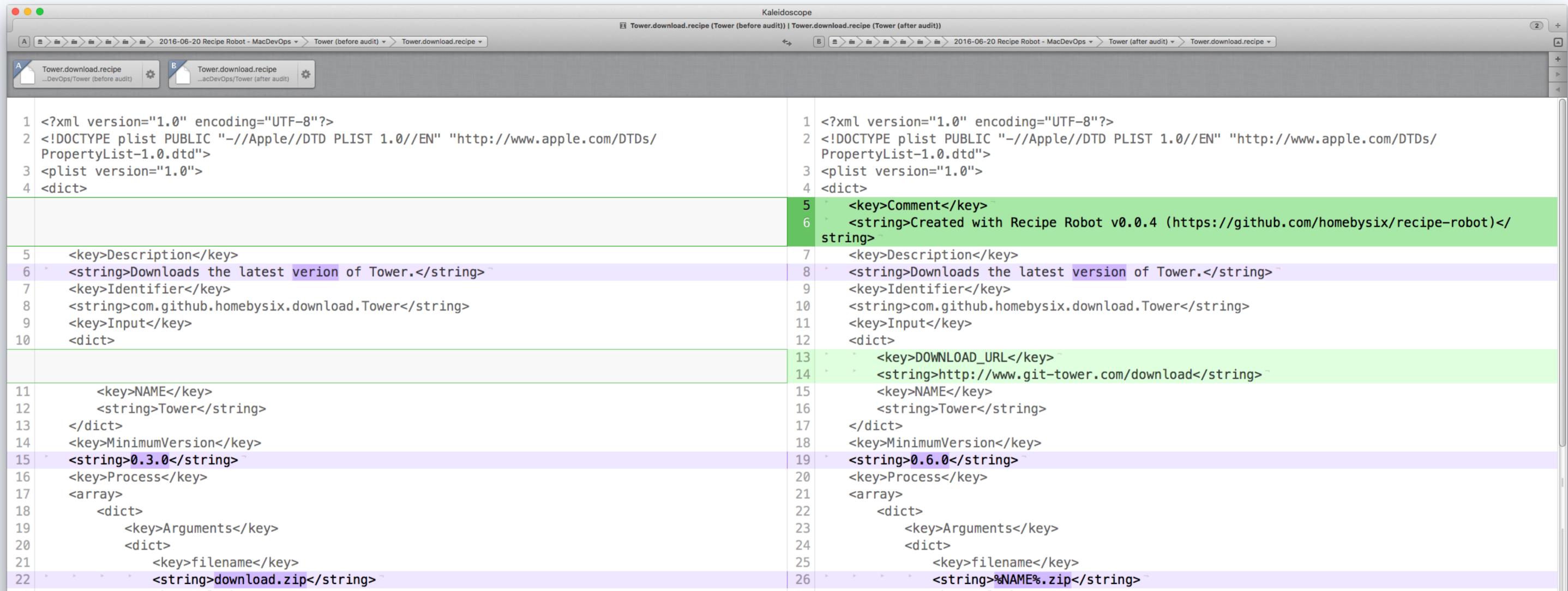


```
-----  
| Welcome to Recipe Robot v1.0.2. |  
-----  
  \  _[]_  
   \ [oo]  
    d-||-b  
     ||  
    _/  \_  
  
Processing http://www.git-tower.com/download ...  
Input path looks like a download URL.  
  Download URL is: http://www.git-tower.com/download  
Downloading file for further inspection...  
  Downloaded to /Users/elliott/Library/Caches/Recipe Robot/2016-06-04_21-24-06_666255/download  
Determining download format...  
Opening downloaded file...  
  Successfully unarchived zip  
Validating app...  
  App seems valid
```

Audit your existing AutoPkg recipes

4. Compare the resulting recipes with the originals.

Use your preferred diff tool for this. (The example below is Kaleidoscope.)



The screenshot shows the Kaleidoscope diff tool comparing two XML files. The left pane shows the original file, and the right pane shows the file after an audit. The diff highlights several changes:

- Line 5: A new key `<key>Comment</key>` is added.
- Line 6: A new string `<string>Created with Recipe Robot v0.0.4 (https://github.com/homebysix/recipe-robot)</string>` is added.
- Line 15: The `<string>0.3.0</string>` is changed to `<string>0.6.0</string>`.
- Line 26: The `<string>download.zip</string>` is changed to `<string>%NAME%.zip</string>`.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Comment</key>
6   <string>Created with Recipe Robot v0.0.4 (https://github.com/homebysix/recipe-robot)</string>
7   <key>Description</key>
8   <string>Downloads the latest version of Tower.</string>
9   <key>Identifier</key>
10  <string>com.github.homebysix.download.Tower</string>
11  <key>Input</key>
12  <dict>
13    <key>DOWNLOAD_URL</key>
14    <string>http://www.git-tower.com/download</string>
15    <key>NAME</key>
16    <string>Tower</string>
17  </dict>
18  <key>MinimumVersion</key>
19  <string>0.6.0</string>
20  <key>Process</key>
21  <array>
22    <dict>
23      <key>Arguments</key>
24      <dict>
25        <key>filename</key>
26        <string>%NAME%.zip</string>
```

Audit your existing AutoPkg recipes

5. Apply the changes you like. Discard the ones you don't.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Description</key>
6   <string>Downloads the latest version of Tower.</string>
7   <key>Identifier</key>
8   <string>com.github.homebysix.download.Tower</string>
9   <key>Input</key>
10  <dict>
11    <key>NAME</key>
12    <string>Tower</string>
13  </dict>
14  <key>MinimumVersion</key>
15  <string>0.3.0</string>
16  <key>Process</key>
17  <array>
18    <dict>
19      <key>Arguments</key>
20      <dict>
21        <key>filename</key>
22        <string>download.zip</string>
23      </dict>
24    </dict>
25  </array>
26  <key>DOWNLOAD_URL</key>
27  <string>http://www.git-tower.com/download</string>
28  <key>NAME</key>
29  <string>Tower</string>
30 </dict>
31 </plist>
32 </xml>
```

The screenshot shows a side-by-side comparison of two XML files in Kaleidoscope. The left pane shows the original recipe, and the right pane shows the recipe after an audit. Changes are highlighted in green and purple. The changes include:

- Line 5: A new key `Comment` is added with the value `Created with Recipe Robot v0.0.4 (https://github.com/homebysix/recipe-robot)`.
- Line 6: The `Description` key is updated from `Downloads the latest version of Tower.` to `Downloads the latest version of Tower.` (no change in text, but highlighted).
- Line 15: The `MinimumVersion` key is updated from `0.3.0` to `0.6.0`.
- Line 22: A new key `DOWNLOAD_URL` is added with the value `http://www.git-tower.com/download`.
- Line 26: The `NAME` key is updated from `Tower` to `%NAME%`.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>Description</key>
6   <string>Downloads the latest version of Tower.</string>
7   <key>Identifier</key>
8   <string>com.github.homebysix.download.Tower</string>
9   <key>Input</key>
10  <dict>
11    <key>NAME</key>
12    <string>Tower</string>
13  </dict>
14  <key>MinimumVersion</key>
15  <string>0.3.0</string>
16  <key>Process</key>
17  <array>
18    <dict>
19      <key>Arguments</key>
20      <dict>
21        <key>filename</key>
22        <string>download.zip</string>
23        <key>url</key>
24        <string>http://www.git-tower.com/download</string>
25      </dict>
26      <key>Processor</key>
27      <string>URLDownloader</string>
28    </dict>
29  </array>
30  <key>Arguments</key>
31  <dict>
32    <key>Comment</key>
33    <string>Created with Recipe Robot v0.0.4 (https://github.com/homebysix/recipe-robot)</string>
34  </dict>
35  <key>DOWNLOAD_URL</key>
36  <string>http://www.git-tower.com/download</string>
37  <key>NAME</key>
38  <string>Tower</string>
39  </dict>
40 </dict>
41 </plist>
42 </xml>
```

✓ Standard description

✓ Made download URL into an input variable.

✓ Standard input variables.

✓ Added missing EndOfCheckPhase.

(continues)

(continued)

```
28     </dict>
29     <dict>
30         <key>Arguments</key>
31         <dict>
32             <key>archive_path</key>
33             <string>%pathname%</string>
34             <key>destination_path</key>
35             <string>%RECIPE_CACHE_DIR%/unzipped</string>
36             <key>purge_destination</key>
37             <true/>
38         </dict>
39         <key>Processor</key>
40         <string>Unarchiver</string>
41     </dict>
42     <dict>
43         <key>Arguments</key>
44         <dict>
45             <key>input_plist_path</key>
46             <string>%RECIPE_CACHE_DIR%/unzipped/Tower.app/Contents/Info.plist</string>
47             <key>plist_version_key</key>
48             <string>CFBundleVersion</string>
49         </dict>
50         <key>Processor</key>
51         <string>Versioner</string>
52     </dict>
53 </array>
54 </dict>
55 </dict>
56 </dict>
57 <dict>
58     <key>Arguments</key>
59     <dict>
60         <key>input_path</key>
61         <string>%RECIPE_CACHE_DIR%/Applications/%NAME%.app</string>
62         <key>requirement</key>
63         <string>anchor apple generic and identifier "com.fournova.Tower2" and
64         (certificate leaf[field.1.2.840.113635.100.6.1.9] /* exists */ or certificate 1[field.
65         1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /
66         * exists */ and certificate leaf[subject.OU] = UN97WY764J)</string>
67     </dict>
68     <key>Processor</key>
69     <string>CodeSignatureVerifier</string>
70 </dict>
71 </dict>
72 </array>
73 </dict>
```

✓ Added CodeSignatureVerifier.

✓ Using correct version key.

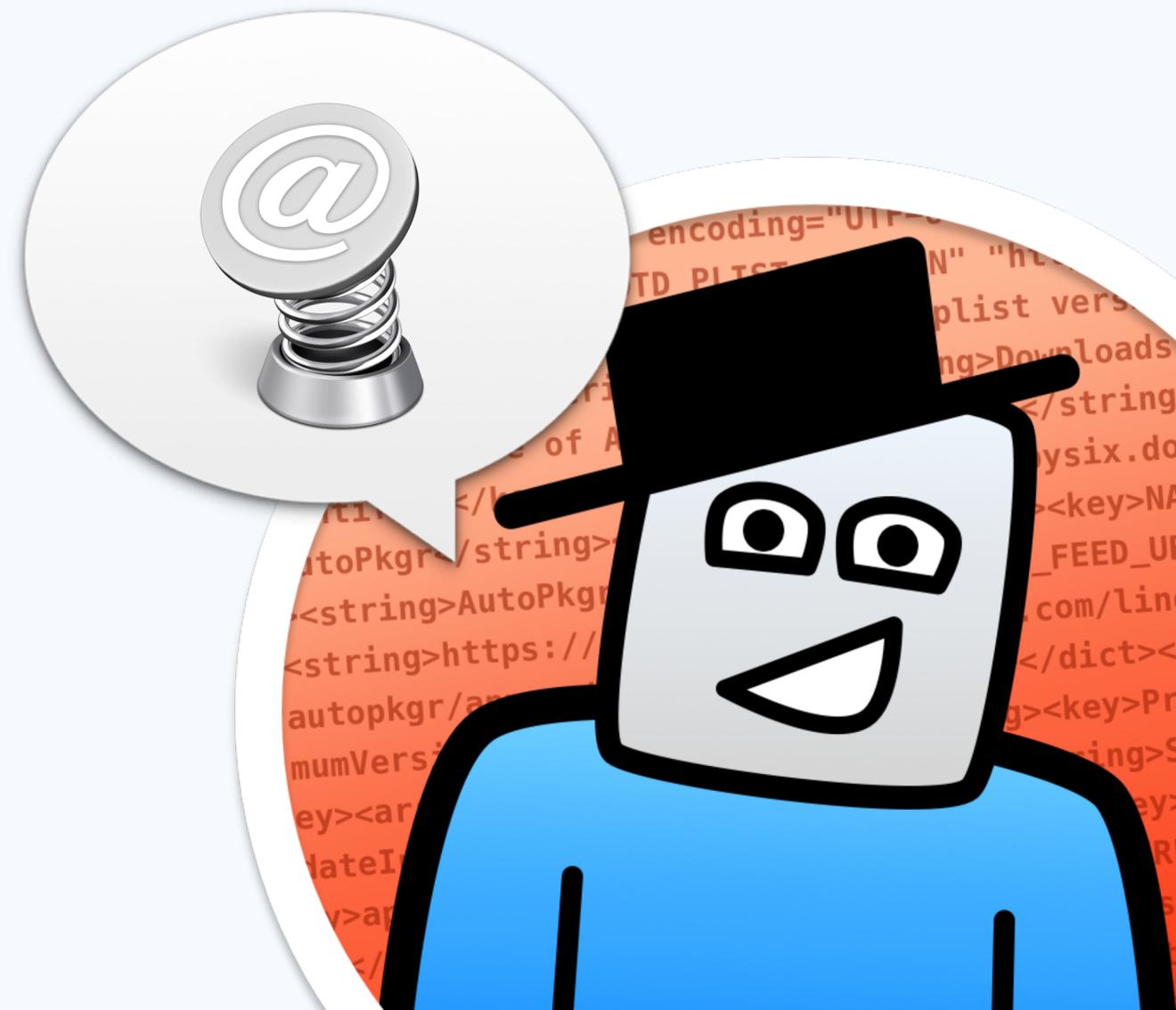
Audit your existing AutoPkg recipes

Recap:

1. Standardize the existing recipes with:
`plutil -convert xml1 *.recipe`
2. Locate and copy the download URL or Sparkle feed.
3. Run that URL through Recipe Robot.
4. Compare the resulting recipes with the originals.
5. Apply the changes you like. Discard the ones you don't.

Augmenting recipes with

URLTextSearcher



Why do we need URLTextSearcher?

C-Command Software DropDMG EagleFiler SpamSieve Blog Forums Search

DropDMG

The easiest way to create and work with Mac disk images (.dmg files).

- Requires: Mac OS X 10.6.8, 10.7, 10.8, 10.9, 10.10, or 10.11 ([older versions](#) available).
- See what's new in version [3.3](#) (May 23, 2016).

"I've been a very happy user of DropDMG. Thanks for all your work."
—Bruce Horn, creator of the Mac Finder

"DropDMG kicks ass!"
—Brian Greenstone, Pangea Software

Download (Free Trial)

Buy DropDMG (\$24)

★★★★★
★★★★☆

Disk images pack entire folders or disks into a single compressed file, either for transport across the Internet or simply for backup. They are Apple's preferred format for distributing Mac software and the only archive format whose contents you can directly access in the Mac Finder—no third-party software required.

Just drag and drop a file or folder into the application window, or click [what to do](#): create a disk image.

<http://c-command.com/downloads/DropDMG-3.3.dmg>

Applications folder, and a [custom volume icon](#).

[Configurations](#) let you reuse common collections of settings, and [temporary options](#) let you easily make one-off changes.

Supports all the major [formats](#), including compressed .dmg files (smaller than Disk Utility's), the new LZSE format, sparsebundle files, .zip and .tar.gz

How does URLTextSearcher work?

```
18 <key>Process</key>
19 <array>
20 <dict>
21 <key>Processor</key>
22 <string>URLTextSearcher</string>
23 <key>Arguments</key>
24 <dict>
25 <key>url</key>
26 <string>http://c-command.com/dropdmg/</string>
27 <key>re_pattern</key>
28 <string>/downloads/DropDMG-([\d\.]+).dmg</string>
29 <key>result_output_var_name</key>
30 <string>version</string>
31 </dict>
32 </dict>
33 <dict>
34 <key>Processor</key>
35 <string>URLDownloader</string>
36 <key>Arguments</key>
37 <dict>
38 <key>url</key>
39 <string>http://c-command.com/downloads/DropDMG-%version%.dmg</string>
40 </dict>
41 </dict>
42 <dict>
43 <key>Processor</key>
44 <string>EndOfCheckPhase</string>
45 </dict>
```

Where to search

What to search for

What to call it

New download URL

Where does Recipe Robot fit in?

1. **Give the current version's download URL to Recipe Robot.**
Ignore the warning you'll probably get about the URL being version-specific.
2. **Edit the download recipe and add a [URLTextSearcher](#) processor with the appropriate arguments.**

Need help with the regular expressions?

Patterns.app · [regex101.com](#) · [regexpr.com](#)

Browser address bar: c-command.com

Navigation: C-Command Software | **DropDMG** | EagleFiler | SpamSieve | Blog | Forums | Search



DropDMG



The easiest way to create and work with Mac disk images (.dmg files).

- Requires: Mac OS X 10.6.8, 10.7, 10.8, 10.9, 10.10, or 10.11 ([older versions](#) available).
- See what's new in version [3.3](#) (May 23, 2016).

Download (Free Trial)

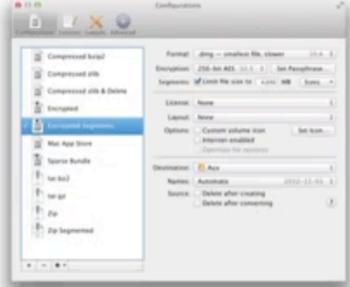
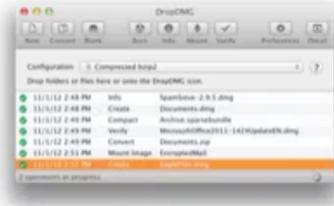
Buy DropDMG (\$24)

About

- Features
- Manual
- Press
- Screenshots
- Support
- Testimonials

"I've been a very happy user of DropDMG. Thanks for all your work."
—Bruce Horn, creator of the Mac Finder

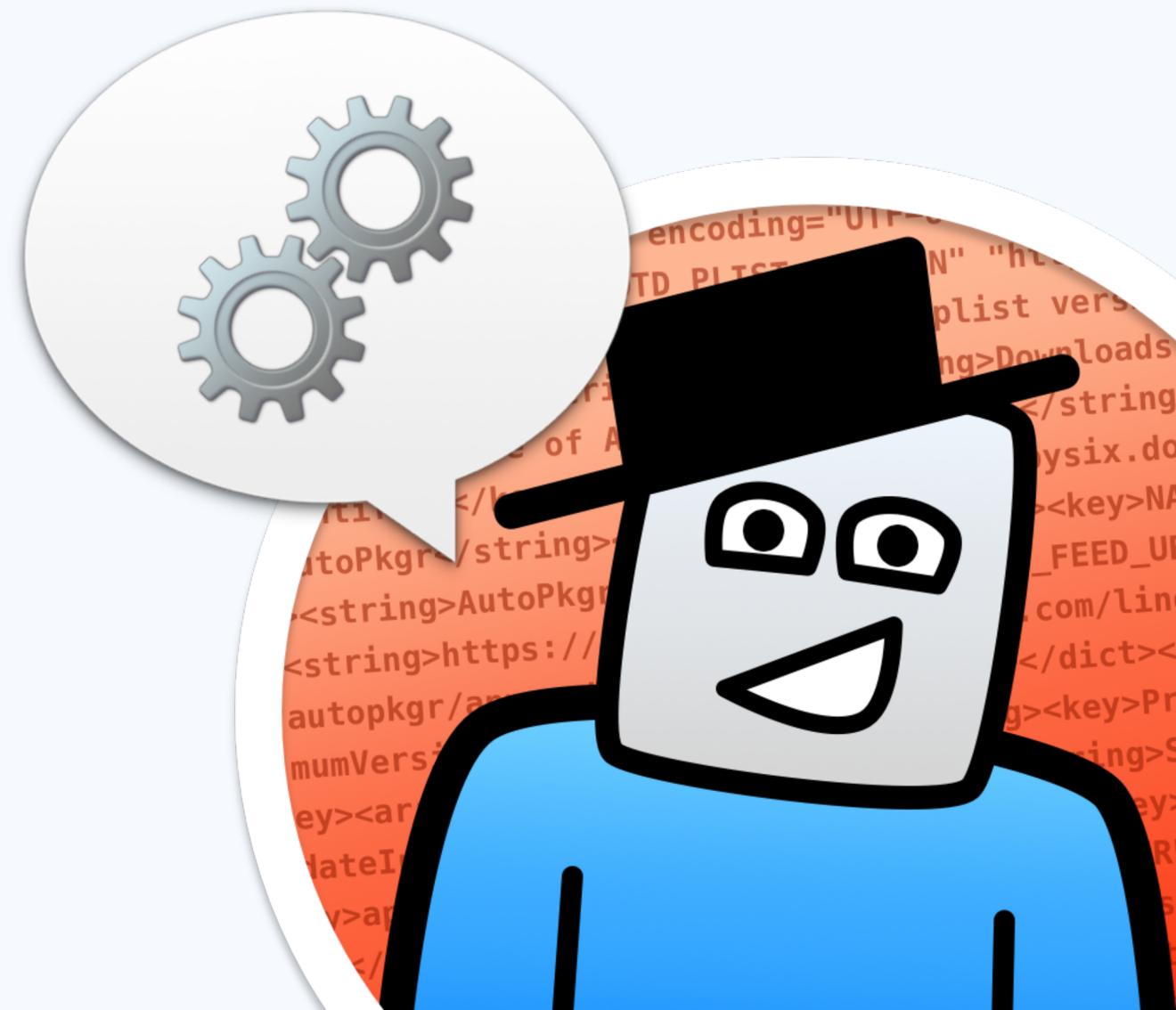
"DropDMG kicks ass!"
—Brian Greenstone, Pangea Software



★★★★★
★★★★★

Using Recipe Robot to help

Create child recipes



Reminder:

Recipe Robot (still) only builds complete recipe sets.



A Manual Workaround

--ignore-existing



Things you may need to tweak in your recipes after changing the ParentRecipe include:

- *Add code signature verification?*
- *Add or adjust versioning?*
- *Modify paths or variable names?*

Building a Child Recipe

- 1. Inspect the existing download recipe for a download URL or Sparkle feed URL.**
- 2. Run that URL through Recipe Robot.**
- 3. Adjust the ParentRecipe of the child recipe.**
- 4. Test and tweak the new child recipe.**

Demo: Building a Child Recipe

There is a `pkg` recipe for `cineSync`, but no `jss` recipe. Let's build one.

```
[Elliot's-MacBook-Pro:~ eliot$ autopkg search cineSync ]
```

Name	Repo	Path
-----	-----	-----
cineSync.download.recipe	gregneagle-recipes	cineSync/cineSync.download.recipe
cineSync.install.recipe	gregneagle-recipes	cineSync/cineSync.install.recipe
cineSync.munki.recipe	gregneagle-recipes	cineSync/cineSync.munki.recipe
cineSync.pkg.recipe	gregneagle-recipes	cineSync/cineSync.pkg.recipe

To add a new recipe repo, use 'autopkg repo-add <repo name>'

```
Elliot's-MacBook-Pro:~ eliot$
```

Use Cases

existing download



new **ds**, **munki**, or **install**

existing pkg



new **jss**, **filewave**, or **lanrev**

existing {foo}



new **{bar}**?

Building a Child Recipe

Recap:

- 1. Inspect the existing download recipe for a download URL or Sparkle feed URL.**
- 2. Run that URL through Recipe Robot.**
- 3. Adjust the ParentRecipe of the child recipe.**
- 4. Test and tweak the new child recipe.**

Recommendations for

Publishing Your Recipes



Should I share my recipes?

Should I share my recipes?

(a) If they would be useful to at least one other person.

Today

1:05 PM

elliottjordan I think I'll write a set of AutoPkg recipes for FantasticApp. Would anybody use them?

 521

 129

 25

 11

Should I share my recipes?

(b) If they are unique.



Should I share my recipes?

(c) If you are willing and able to maintain them.



What do I get out of it?

What do I get out of it?

Your recipes appear in [autopkg search](#) results.

```
$ autopkg search FantasticApp
```

Name	Repo	Path
-----	-----	-----
FantasticApp.download.recipe	homebysix-recipes	FantasticApp/FantasticApp.download.recipe
FantasticApp.install.recipe	homebysix-recipes	FantasticApp/FantasticApp.install.recipe
FantasticApp.munki.recipe	homebysix-recipes	FantasticApp/FantasticApp.munki.recipe
FantasticApp.pkg.recipe	homebysix-recipes	FantasticApp/FantasticApp.pkg.recipe

To add a new recipe repo, use 'autopkg repo-add <repo name>'

What do I get out of it?

Peer review and feedback.

The screenshot shows a web browser window with the GitHub website. The browser's address bar shows 'GitHub, Inc.' and the page title is 'Royal TSX Fails to download #56'. The navigation bar includes 'Code', 'Issues 5', 'Pull requests 0', 'Pulse', and 'Graphs'. The issue is marked as 'Closed' and was opened by 'seansgm' on May 4 with 2 comments. The issue content shows a comment from 'seansgm' reporting an error: 'Processor: URLTextSearcher: Error: No match found on URL: http://www.royalapplications.com/ts/osx/download'. A second comment from 'seansgm' states 'Created a pull request: 225c5ed'. A third comment from 'homebysix' says 'Thanks!'. The right sidebar shows 'Labels: None yet', 'Milestone: No milestone', 'Assignees: No one assigned', and '2 participants' with avatars for 'seansgm' and 'homebysix'. At the bottom, a red circle with a slash indicates the issue was closed by 'homebysix' on May 4.

Code Issues 5 Pull requests 0 Pulse Graphs

Royal TSX Fails to download #56

Closed seansgm opened this issue on May 4 · 2 comments

Labels
None yet

Milestone
No milestone

Assignees
No one assigned

2 participants
seansgm homebysix

seansgm commented on May 4

```
Processor: URLTextSearcher: Error: No match found on URL:  
http://www.royalapplications.com/ts/osx/download
```

seansgm commented on May 4

Created a pull request: [225c5ed](#)

homebysix commented on May 4 autopkg member

Thanks!

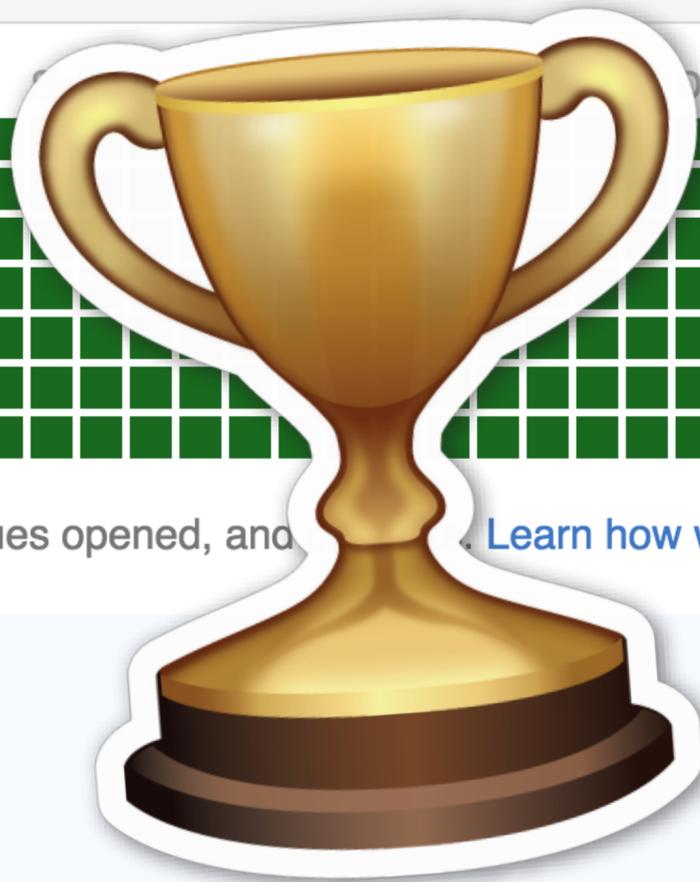
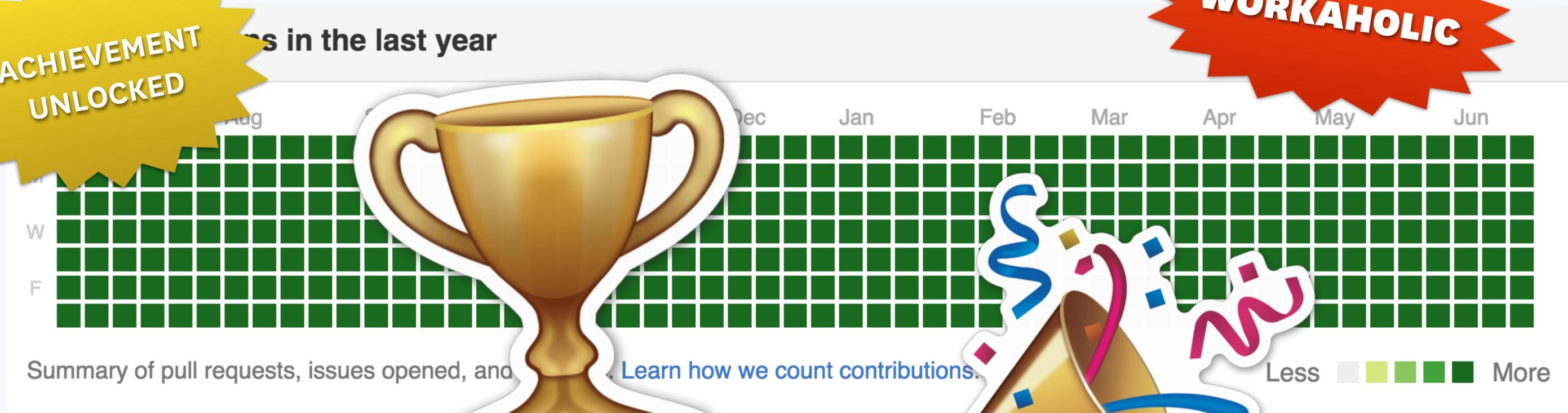
homebysix closed this on May 4

What do I get out of it?

An active GitHub profile. (Great resume builder!)

**ACHIEVEMENT
UNLOCKED**

WORKAHOLIC



What do I get out of it?

High fives from your fellow Mac admins.



What do I get out of it?

High fives from your fellow Mac admins.



What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** `com.github.YourUsername.RecipeType.AppName` (e.g. `com.github.homebysix.download.coconutBattery`).
 - ❌ `A3723087-3AB1-48D5-8684-3C767559D962`
 - ❌ `com.github.homebysix.coconutBattery.download`
 - ❌ `com.apple.download.coconutBattery`

What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** `com.github.YourUsername.RecipeType.AppName` (e.g. `com.github.homebysix.download.coconutBattery`).
- **Overrideability:** Allow administrators to customize things.

- ✓ Munki category and catalog

- ✓ Munki/JSS description

- ✓ Major version

What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** com.github.YourUsername.RecipeType.AppName (e.g. com.github.homebysix.download.coconutBattery).
- **Overrideability:** Allow administrators to customize results.
- **Versioning:** Even if your recipe doesn't need a version, child recipes might someday.

What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** com.github.YourUsername.RecipeType.AppName (e.g. com.github.homebysix.download.coconutBattery).
- **Overrideability:** Allow administrators to customize results.
- **Versioning:** Even if your recipe doesn't need a version, child recipes might someday.
- **Behavior:** Import into Munki "testing" catalog.

What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** `com.github.YourUsername.RecipeType.AppName` (e.g. `com.github.homebysix.download.coconutBattery`).
- **Overrideability:** Allow administrators to customize results.
- **Versioning:** Even if your recipe doesn't need a version, child recipes might someday.
- **Behavior:** Import into Munki "testing" catalog.
- **Linting:** `plutil -lint *.recipe`

What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** `com.github.YourUsername.RecipeType.AppName` (e.g. `com.github.homebysix.download.coconutBattery`).
- **Overrideability:** Allow administrators to customize results.
- **Versioning:** Even if your recipe doesn't need a version, child recipes might someday.
- **Behavior:** Import into Munki "testing" catalog.
- **Linting:** `plutil -lint *.recipe`
- **Avoid SourceForge:** There are often better sources.

UPTIME CLOUD MONITOR

CPU Network Storage Load

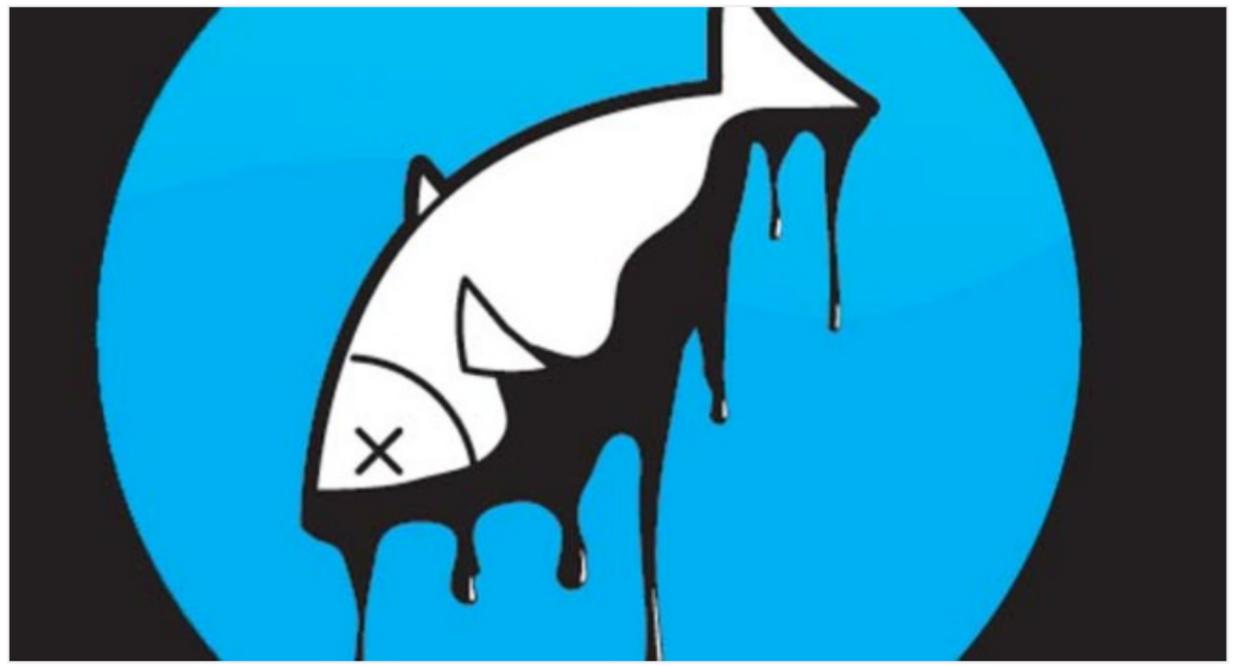
START FOR FREE

IDERA
NEVER. SLOW. DOWN.

Business

SourceForge sorry for adware, promises only opt-in in future

Download site slaps self after GIMP grump, outlines meta-adware loopholes



3 Jun 2015 at 04:56, [Simon Sharwood](#)

Reddit Twitter Facebook 19 LinkedIn 4

Software download service SourceForge has changed its adware-insertion policies after [earning](#) users' ire for wrapping popular FOSS image-wrangling app The GIMP in an adware-riddled downloader.

The Best NetFlow Analyzer

Monitors Your Network and Internet Bandwidth - Download Free Version.

Green button with a white right-pointing arrow.

Most read

 Who would code a self-destruct feature into their own web browser? Oh, hello, Apple

 Who wants a quad-core 4.2GHz, 64GB, 5TB SSD RAID 10 ... laptop?

 Winning Underhand C Contest code silently tricks nuke inspectors

 Mystery hacker hijacks

We've moved! You can now read the latest and greatest on Mac adware and malware at Malwarebytes. >

Java now installing adware

Published March 4th, 2015 at 11:34 AM EST , modified March 4th, 2015 at 11:34 AM EST



Rich Trouton, a Mac systems administrator who runs the Der Flounder blog, discovered yesterday that a Java installer is installing adware, in the form of the Ask Toolbar. (He first wrote about it on JAMF Nation, but has published additional information in his Der Flounder post today.) Fortunately, in the course of trying to duplicate his findings, it appears that this installer is a bit finicky, and may not always install the

toolbar properly. I had a slight bit of trouble finding the troublesome installer at first. My search initially took me to Oracle's site, where I downloaded Java 8 Update 40 and found that it was just a simple installer package, with no nasty hitchhikers. My second step – to Java.com – hit paydirt, though, with the Mac installer downloaded from that site being the application described by Trouton.



On running that installer, I initially saw exactly what Trouton described. At one point in the install process, I was asked whether I wanted to install the Search App by Ask.



NEW
Malwarebytes ANTI-MALWARE
 FOR MAC

Removes adware and malware
 and revives your Mac

DOWNLOAD FOR FREE

OFFICIAL SECURITY BLOG

Malwarebytes UNPACKED

Home Authors Videos Scams About Us Archives + Categories +

Has MacUpdate fallen to the adware plague?

NOVEMBER 2, 2015 | BY THOMAS REED



A Twitter user named [Ciro Urdaneta](#) made a concerning discovery on Friday:

 **Ciro Urdaneta** @curdaneta · Oct 30
 @macupdate this is what I got from your Skype Installer.dmg Bitdefender warned me #NotCool #malware

In other words, it seems that MacUpdate, long considered to be one of the only remaining trustworthy download aggregation sites for the Mac, has succumbed to the same plague that has ruined most of the others: adware.

Following Mr. Urdaneta's hints, I sought out the Skype page on the MacUpdate site and downloaded the app. The result was a file named Skype Installer.dmg, which seems legit on first glance. However, opening this disk image file results in a MacUpdate installer, very similar to the adware-riddled custom installers used by sites like Download.com and Softonic.



MacUpdate Installer

Trojan.DNSChanger circumvents Powershell restrictions

Was Mac OS X really the most vulnerable in 2015?

Tweets

Follow @Malwarebytes

 **Help Net Security** @helpnetsecurity 11h
 Fake Amazon survey-for-money offer leads to account compromise - [bit.ly/1Pn51dd](#) - @paperghost @Malwarebytes [pic.twitter.com/uArlPQKWhB](#)
 Retweeted by Malwarebytes
 Show Photo

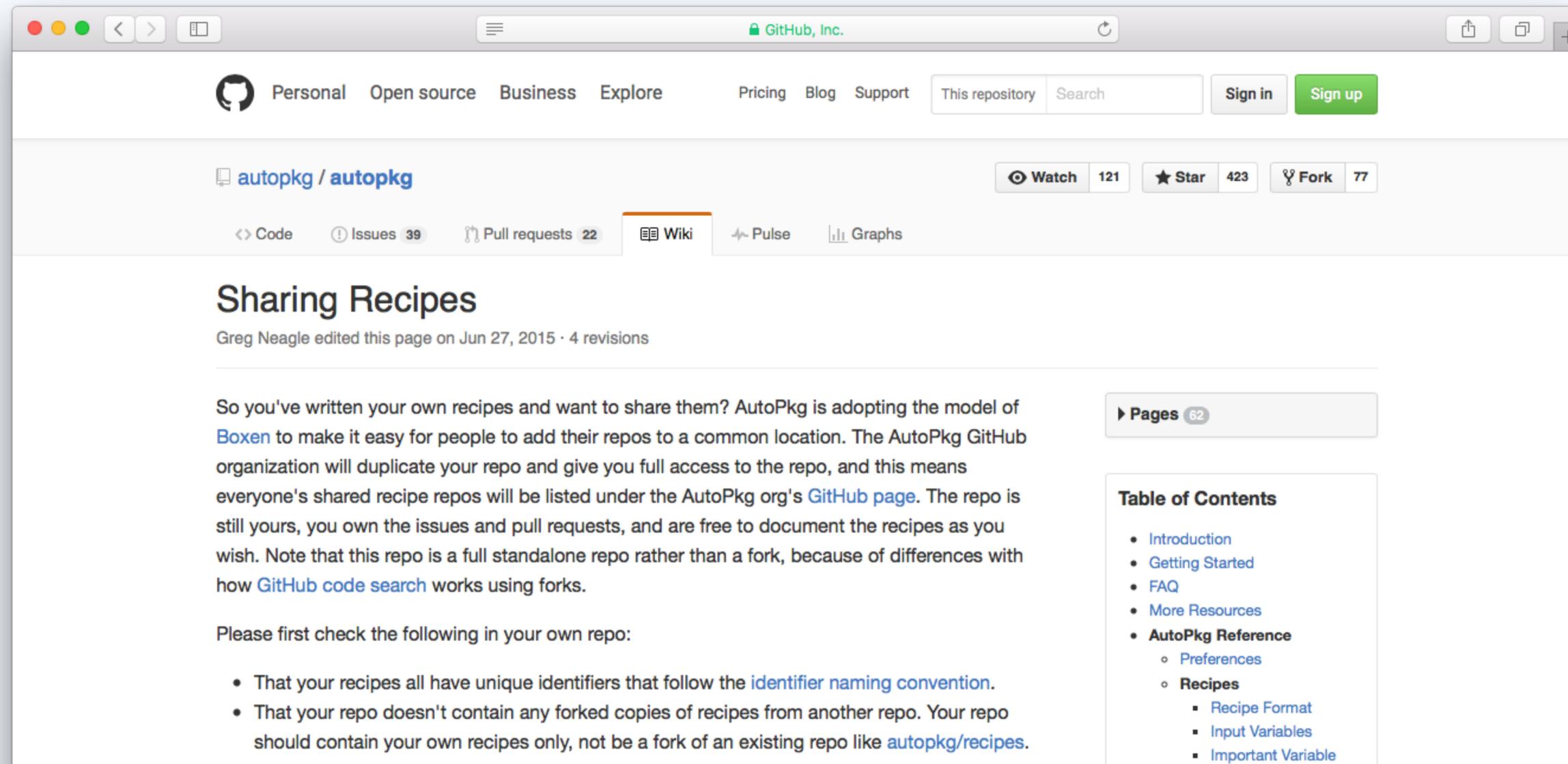
 **Techworld** @techworldnews 11h
 TMZ.com is latest victim of #malware advertising campaign: [bit.ly/23LdHnt](#) @Malwarebytes [pic.twitter.com/Rjfn6SbkpA](#)
 Retweeted by Malwarebytes

Users are told to download.

What kind of cleanup is necessary before I share my recipes?

- **Identifiers:** `com.github.YourUsername.RecipeType.AppName` (e.g. `com.github.homebysix.download.coconutBattery`).
- **Overrideability:** Allow administrators to customize results.
- **Versioning:** Even if your recipe doesn't need a version, child recipes might someday.
- **Behavior:** Import into Munki "testing" catalog.
- **Linting:** `plutil -lint *.recipe`
- **Avoid SourceForge:** There are often better sources.
- **Optimization:** `EndOfCheckPhase`, `PathDeleter`.

How do I publish my recipes?



The screenshot shows a browser window displaying the GitHub repository page for 'autopkg/autopkg'. The page title is 'Sharing Recipes', edited by Greg Neagle on Jun 27, 2015. The main content explains the 'Boxen' model for sharing recipes and lists requirements for contributors. A table of contents on the right lists sections like 'Introduction', 'Getting Started', 'FAQ', 'More Resources', and 'AutoPkg Reference' with sub-sections like 'Preferences', 'Recipes', 'Recipe Format', 'Input Variables', and 'Important Variable'.

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

autopkg / autopkg Watch 121 Star 423 Fork 77

Code Issues 39 Pull requests 22 Wiki Pulse Graphs

Sharing Recipes

Greg Neagle edited this page on Jun 27, 2015 · 4 revisions

So you've written your own recipes and want to share them? AutoPkg is adopting the model of [Boxen](#) to make it easy for people to add their repos to a common location. The AutoPkg GitHub organization will duplicate your repo and give you full access to the repo, and this means everyone's shared recipe repos will be listed under the AutoPkg org's [GitHub page](#). The repo is still yours, you own the issues and pull requests, and are free to document the recipes as you wish. Note that this repo is a full standalone repo rather than a fork, because of differences with how [GitHub code search](#) works using forks.

Please first check the following in your own repo:

- That your recipes all have unique identifiers that follow the [identifier naming convention](#).
- That your repo doesn't contain any forked copies of recipes from another repo. Your repo should contain your own recipes only, not be a fork of an existing repo like [autopkg/recipes](#).

Pages 62

Table of Contents

- [Introduction](#)
- [Getting Started](#)
- [FAQ](#)
- [More Resources](#)
- **AutoPkg Reference**
 - [Preferences](#)
 - **Recipes**
 - [Recipe Format](#)
 - [Input Variables](#)
 - [Important Variable](#)

<https://github.com/autopkg/autopkg/wiki/Sharing-Recipes>

2. We'll duplicate the repo, give it a name like "username-recipes," and give you full write access to the repo. Indicate if whether you'd like a different name for the recipes (for example, if your repo has a specific collection or "category" of software).

- [Recipe-writing Guidelines](#)
- [Using CodeSignatureVerifier](#)

Be A Good Recipe Neighbor

Always be testing.



- Automate your testing runs with AutoPkgr.
- Or with a super simple one-liner:

```
find homebysix-recipes \  
  -iname "*.recipe" \  
  -exec autopkg run -v "{}" \;
```
- Or with a more intelligent script.
- At least test the *download* recipes regularly. Those are the ones most likely to fail.

Be A Good Recipe Neighbor

Collaborate.



- Incoming issues.
- Incoming pull requests.
- Outgoing issues and pull requests.
- MacAdmins.org Slack → #autopkg.

Be A Good Recipe Neighbor

Make a *commitment* to good commits.



- Yes, people read your commit messages.
- Be brief, but descriptive.
- Make commits after each discrete change.

Be A Good Recipe Neighbor

Avoid a few party-fouls.

- 🚫 Don't change recipe identifiers. (Guilty!)
- 🚫 Don't push before testing first.
- 🚫 Don't introduce plist syntax errors.

Be A Good Recipe Neighbor

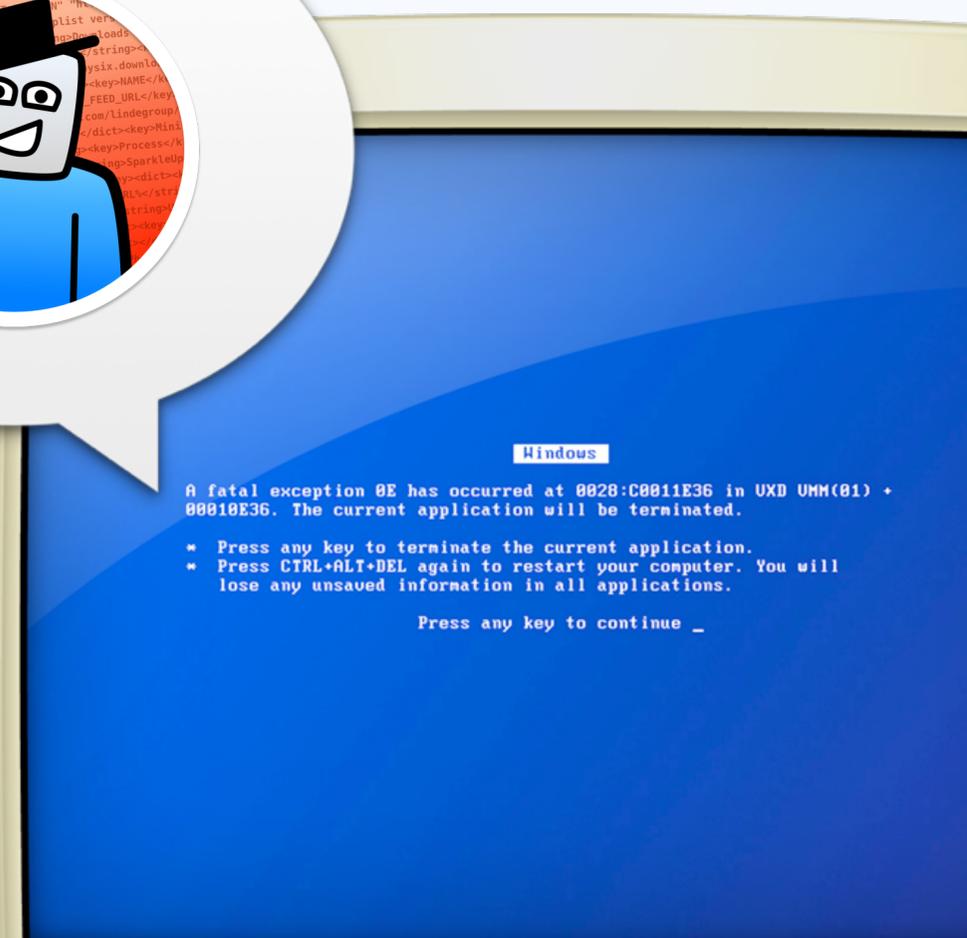
Get secure. Stay secure.



- FileVault on your Mac.
- Two factor authentication.
- Good passwords.
- Ditto for your email too, not just GitHub.

Troubleshooting

Sometimes the robot explodes.



```
Windows
A fatal exception 0E has occurred at 0020:C0011E36 in UXD UHM(01) +
00010E36. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will
lose any unsaved information in all applications.
Press any key to continue _
```

Troubleshooting Tips and Methods

- **Run with `--verbose`.**

Check for warnings or errors. I run it in verbose mode every time.

- **Try again with different input.**

Possibilities: Sparkle feed, download URL, GitHub, BitBucket, or SourceForge URL, zip, dmg, pkg, app.

- **Download the app yourself, and have a look at it.**

Maybe it's nested inside a folder (Audacity).

Maybe there are multiple apps in the dmg (Airfoil).

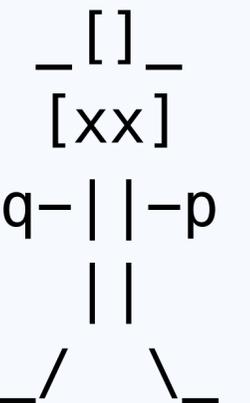
Maybe it's a binary instead of an app (Vagrant).

- **Keep the cache with `--keep-cache`.**

Allows you to see all the unpacking Recipe Robot tried to do at `~/Library/Caches/Recipe Robot`.

- **Open a GitHub issue, include your input.**

We'll have a look and see whether we can improve Recipe Robot's behavior.



Thank you!



Elliot Jordan
@homebysix



Shea Craig
@shea_craig

Questions?