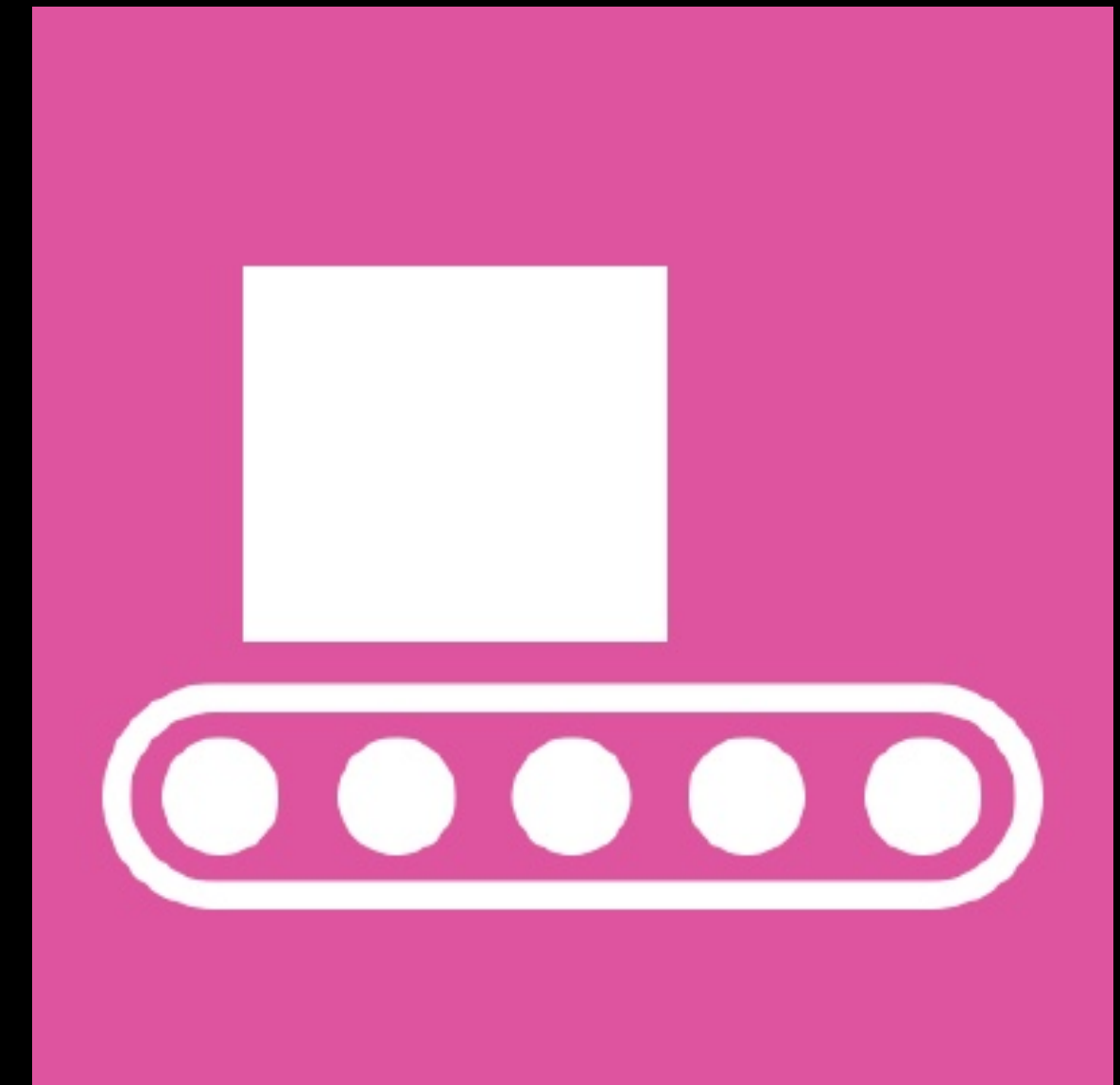


# AutoPkg in Depth

Understanding, Selecting, Creating, and  
Maintaining Recipes

Feedback: <https://bit.ly/psumac2023-102>



**Elliot Jordan**

Netflix

 *homebysix*

 *elliotjordan*

 *homebysix*

**Anthony Reimer**

University of Calgary

 *@AnthonyReimer@mastodon.social*

 *jazzace*

 *jazzace*

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**

# Our Purpose

## Basic:

- Choosing/Understanding Recipes

## Intermediate/Advanced:

- Creating/Maintaining Recipes
- Best Practices

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**

# Outline

AutoPkg prerequisites and basics

Recipe reading and literacy

Running recipes via overrides

Specific recipe types

Creating new recipes

Testing your recipes

Sharing your recipes

Advanced topics (you tell us!)

Wrap up

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**

# AutoPkg Prerequisites and Basics

# Installing AutoPkg

- Requires AutoPkg, Python3, Git
  - ▶ *[github.com/autopkg/autopkg/releases](https://github.com/autopkg/autopkg/releases)*
  - ▶ *Xcode Command Line Tools:*  
*[developer.apple.com/download/](https://developer.apple.com/download/)*
- Automated installation:
  - ▶ *AutoPkgSetup – Graham Pugh*  
*[github.com/grahampugh/AutoPkgSetup](https://github.com/grahampugh/AutoPkgSetup)*
  - ▶ *AutoPkg*

# Standard Workflow

1. Search for recipe(s)

2. Add recipe repo(s)

▶ *Include repos of parent recipes as needed*

3. Create override(s)

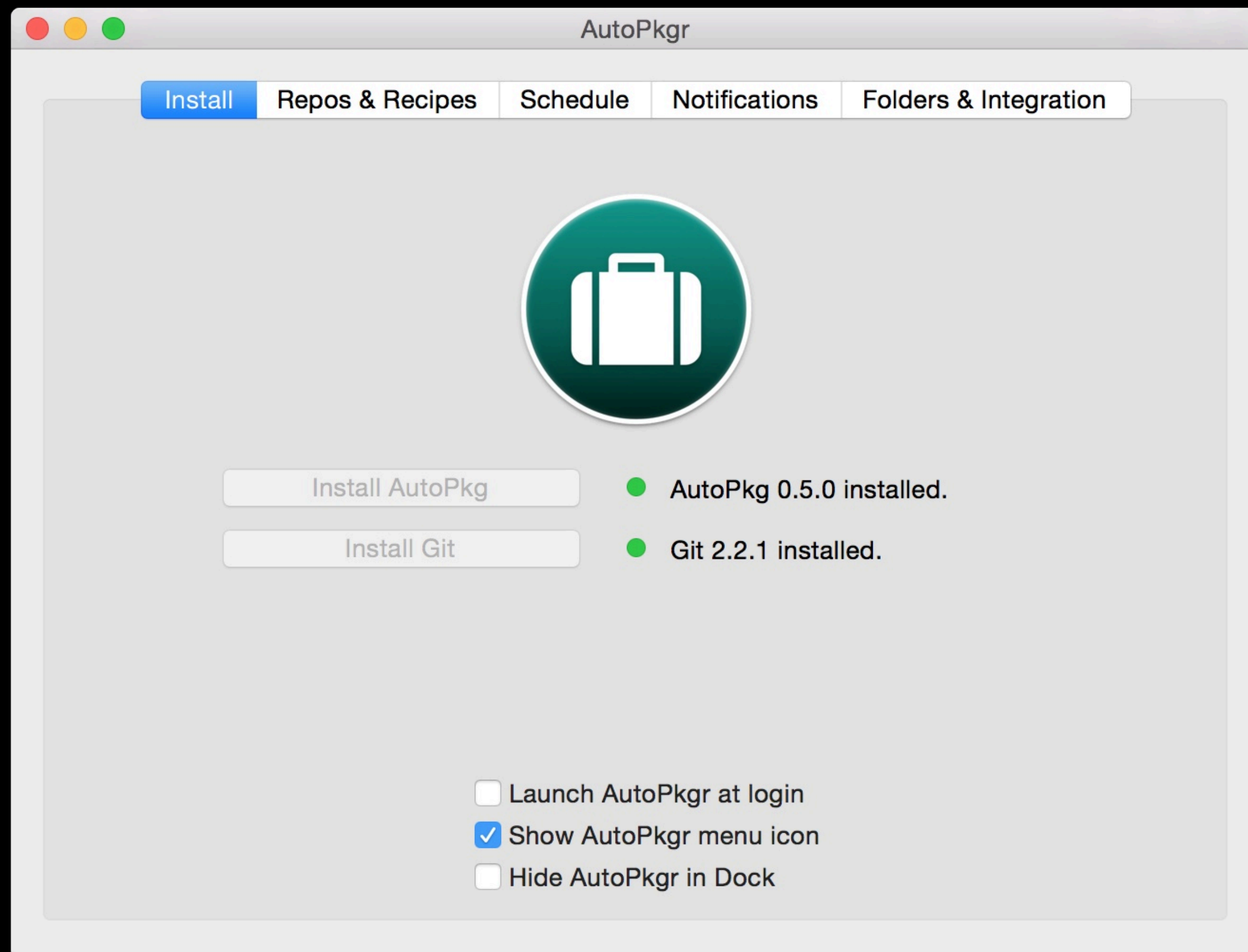
4. Run the override(s)

5. Maintain

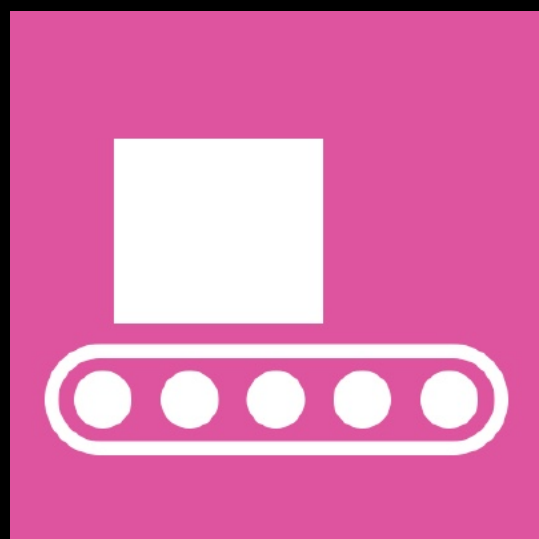
▶ *Update repos and override trust info*



# AutoPkgr



- GUI front end for AutoPkg
  - ▶ [\*github.com/lindeggroup/autopkg\*](https://github.com/lindeggroup/autopkg)



*command line*



User interface

Runs AutoPkg

Assisted Initial Setup

Access to all run options

Processor list & info

Includes Scheduled Runs

Includes Notifications

Supports Trust verification

In Active Development



*GUI*



*limited*



*limited*



# Recipe Reading & Literacy

## Self Saucing Fudge Dessert

*When Joan serves this to company, she puts a scoop of ice cream on the pudding base before topping it with the sauce.*

### Pudding Base

1 cup	flour	250 mL
2 tsp.	baking powder	10 mL
¼ tsp.	salt	1 mL
½ cup	sugar	125 mL
3 tbsp.	cocoa	50 mL
½ cup	chopped nuts	125 mL
½ cup	milk	125 mL
1 tsp.	vanilla	5 mL
2 tbsp.	melted butter or margarine	25 mL

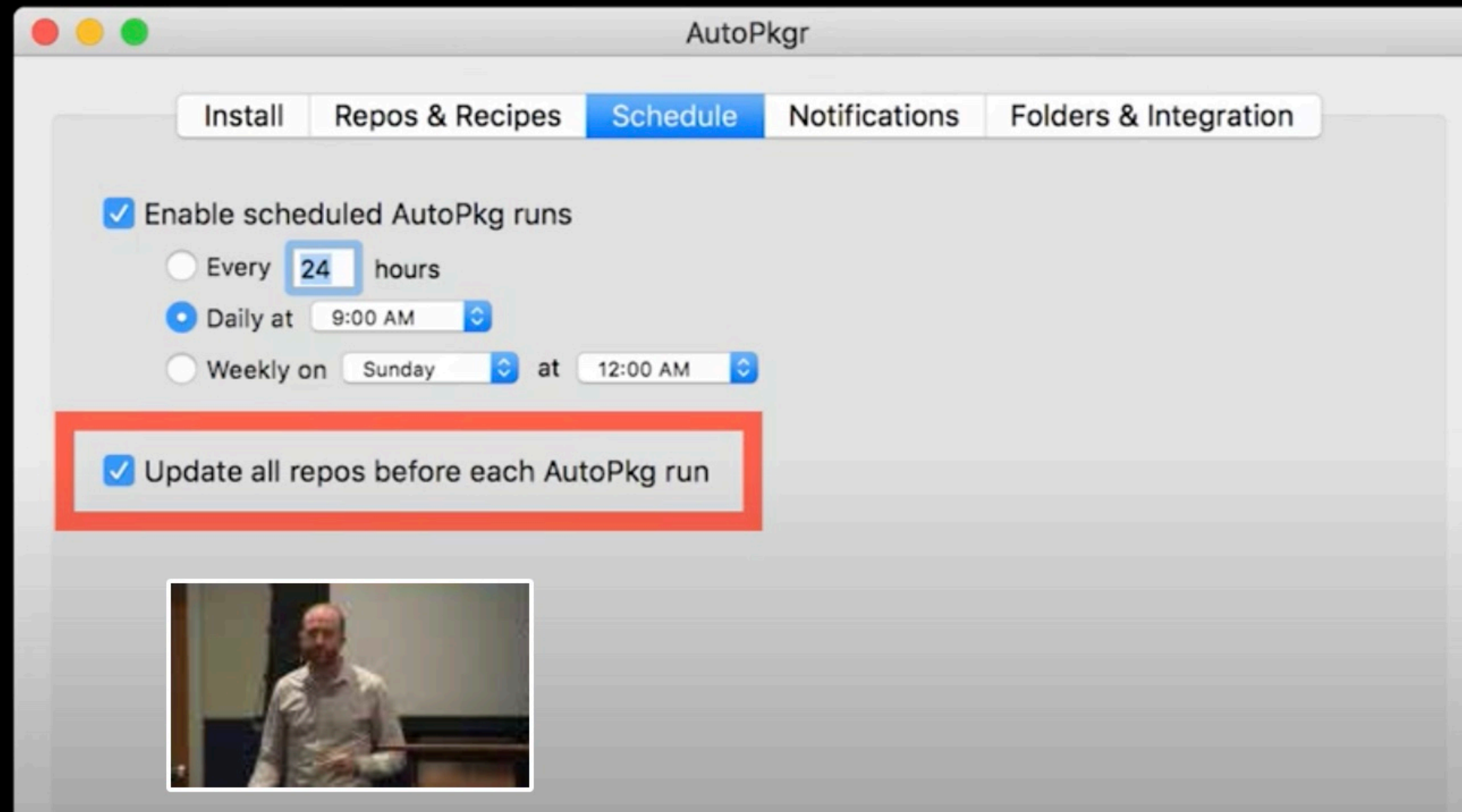
### Sauce

4 tbsp.	cocoa	65 mL
½ cup	sugar	125 mL
2 cups	hot water	500 mL

Sift into a two quart (litre) casserole, flour, baking powder, salt, ½ cup (125 mL) of sugar and 3 tablespoons (50 mL) of cocoa. Stir in the nuts. Make a well in the centre and add milk, vanilla and melted butter. Stir lightly and smooth evenly over bottom of the casserole.

Mix ½ cup (125 mL) of sugar and 4 tablespoons (65 mL) of cocoa. Sprinkle mixture over pudding base. Pour hot water over cocoa sugar mixture. Bake at 350°F (180°C) for 30 minutes. The batter rises and a rich sauce forms under it. Serve hot.

# #7: Trust Without Verification



*How (Not) To Do Bad Things With AutoPkg – Elliot Jordan*  
<https://youtube.com/watch?v=LnvQmLcDF8w>

Name

Description

Ingredients

Instructions

## Self Saucing Fudge Dessert

*When Joan serves this to company, she puts a scoop of ice cream on the pudding base before topping it with the sauce.*

### Pudding Base

1 cup	flour	
2 tsp.	baking powder	250 mL
¼ tsp.	salt	10 mL
½ cup	sugar	1 mL
3 tbsp.	cocoa	125 mL
½ cup	chopped nuts	50 mL
½ cup	milk	125 mL
1 tsp.	vanilla	125 mL
2 tbsp.	melted butter or margarine	5 mL
		25 mL

### Sauce

4 tbsp.	cocoa	
½ cup	sugar	65 mL
2 cups	hot water	125 mL
		500 mL

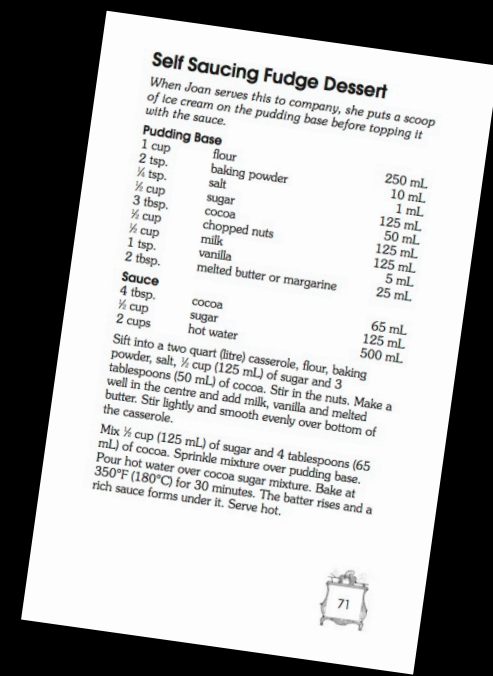
Sift into a two quart (litre) casserole, flour, baking powder, salt, ½ cup (125 mL) of sugar and 3 tablespoons (50 mL) of cocoa. Stir in the nuts. Make a well in the centre and add milk, vanilla and melted butter. Stir lightly and smooth evenly over bottom of the casserole.

Mix ½ cup (125 mL) of sugar and 4 tablespoons (65 mL) of cocoa. Sprinkle mixture over pudding base. Pour hot water over cocoa sugar mixture. Bake at 350°F (180°C) for 30 minutes. The batter rises and a rich sauce forms under it. Serve hot.

Identifier  
Description

Input

Process



# About AutoPkg Recipes

- Plist (XML) or YAML
- Series of steps (Processors)
- Can leverage other recipes ("Parent-Child")
- Best practice is to create a recipe chain rather than an omnibus (monolithic) recipe



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Description</key>
    <string>This recipe does nothing.</string>
    <key>Identifier</key>
    <string>com.github.yourgithubusername.download.nothing</string>
    <key>Input</key>
    <dict>
      <key>NAME</key>
      <string>nothing</string>
    </dict>
    <key>MiniumumVersion</key>
    <string>1.0</string>
    <key>Process</key>
    <array>
      <dict>
        <key>Arguments</key>
        <dict>
          <key>Argument1</key>
          <string>Value1</string>
          <key>Argument2</key>
          <string>Value2</string>
        </dict>
        <key>Processor</key>
        <string>EndOfCheckPhase</string>
      </dict>
    </array>
  </dict>
</plist>
```

Doc Type Def

Description

Identifier

Input

Minimum Version (of AutoPkg)

Processor(s)

Process

Description: This recipe does nothing.

Identifier: com.github.yourgithubusername.download.nothing

Input:

NAME: nothing

MinimumVersion: '2.3'

Process:

– Arguments:

Argument1: Value1

Argument2: Value2

Processor: ProcessorName

# Plist

# YAML



Supported by AutoPkg



(2.3+)



Supported by AutoPkg\_r



9000

Recipes Available

~500

Moderate

Readability

Easy



Can chain with other format



autopkg / grahampugh-recipesPublic

Edit Pins

Watch14

Fork35

Starred63

<> Code

Issues8

Pull requests3

Actions

Projects

Wiki

Security

Insights

main


2 branches

0 tags

Go to file

















Add file

<> Code

 grahampugh

some fixes

3edcda3 3 weeks ago 1,039 commits

	.vscode	Changes to enable policy deletion	3 years ago
	AdobeCreativeCloud	remove force pkg build	2 months ago
	AdobeLicensingToolkit	remove force pkg build	2 months ago
	Affinity	switch to yaml recipes (#63)	2 years ago
	AmazonCorrettoJDK	switch to yaml recipes (#63)	2 years ago
	AppleHPPrinterDrivers	remove StopProcessingIf checks	2 years ago
	AppleJava6	Update Python path in processors	10 months ago
	CLCGenomicsWorkbench	Add user agent	3 months ago
	COMSOL	switch to yaml recipes (#63)	2 years ago
	ChemDraw	remove force pkg build	2 months ago
	CiscoAnyConnect	switch to yaml recipes (#63)	2 years ago
	CitrixWorkspace	change parent	2 months ago
	CommonProcessors	more userinstall recipes	last month
	CotEditor	tidy up jamf recipes and deprecate jss recipes	last year
	DJView	switch to yaml recipes (#63)	2 years ago
	DataWarrior	PathDeleter	last year

About

AutoPkg recipes

Readme

Apache-2.0 license

63 stars

14 watching

35 forks

Report repository








Releases





No releases published

Packages

No packages published

Contributors33





+ 22 contributors

<https://github.com/autopkg/grahampugh-recipes>



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Description</key>
  <string>This recipe calls a recipe that does nothing.</string>
  <key>Identifier</key>
  <string>com.github.yourgithubusername.pkg.nothing</string>
  <key>Input</key>
  <dict>
    <key>NAME</key>
    <string>nothing</string>
  </dict>
  <key>MiniumumVersion</key>
  <string>1.0</string>
  <key>ParentRecipe</key>
  <string>com.github.yourgithubusername.download.nothing</string>
  <key>Process</key>
  <array>
    <dict>
      <key>Arguments</key>
      <dict>
        <key>Argument1</key>
        <string>Value1</string>
      </dict>
      <key>Processor</key>
      <string>AppPkgCreator</string>
    </dict>
  </array>
</dict>
</plist>
```

Description

Identifier

Input

MinimumVersion

ParentRecipe

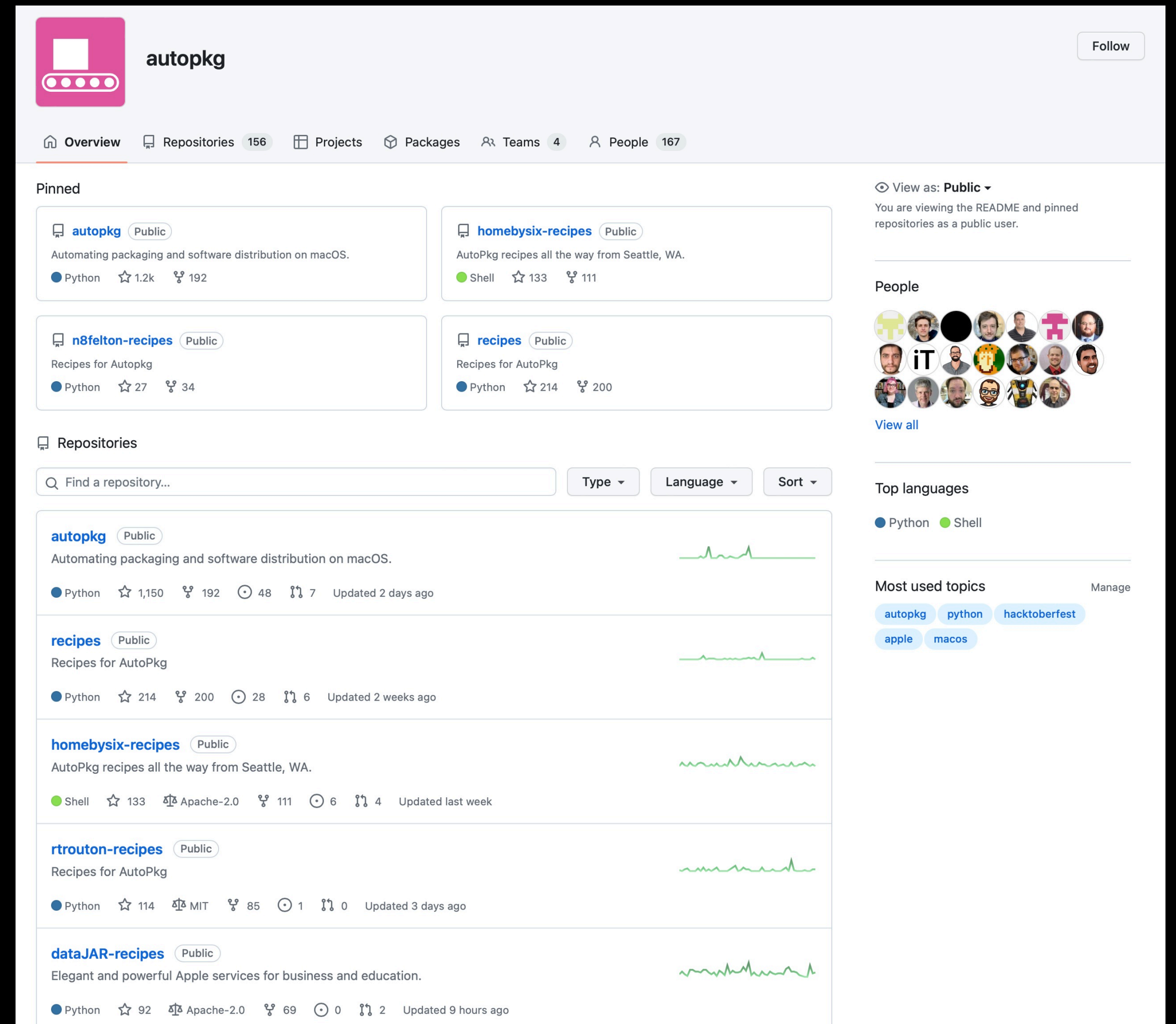
Processor(s)

Process

# AutoPkg org

github.com/autopkg

- AutoPkg code base, core processors, recipes repo
- *yourname*-recipes repo
- Pull Request (fix/improve)
- Standards (see Wiki)



The screenshot shows the GitHub organization page for 'autopkg'. The header includes the organization name and a 'Follow' button. The navigation bar lists 'Overview', 'Repositories' (156), 'Projects', 'Packages', 'Teams' (4), and 'People' (167). The 'Pinned' section displays four repositories: 'autopkg' (Python, 1.2k stars, 192 forks), 'homebysix-recipes' (Shell, 133 stars, 111 forks), 'n8felton-recipes' (Python, 27 stars, 34 forks), and 'recipes' (Python, 214 stars, 200 forks). The 'Repositories' section features a search bar and filters for 'Type', 'Language', and 'Sort'. It lists several repositories with their descriptions, languages, star/fork counts, and update times. The right sidebar shows 'View as: Public', a note about viewing the README, a 'People' section with member avatars, 'Top languages' (Python, Shell), and 'Most used topics' (autopkg, python, hacktoberfest, apple, macos).

autopkg

Follow

Overview Repositories 156 Projects Packages Teams 4 People 167

Pinned

- autopkg** (Public)  
Automating packaging and software distribution on macOS.  
Python 1.2k 192
- homebysix-recipes** (Public)  
AutoPkg recipes all the way from Seattle, WA.  
Shell 133 111
- n8felton-recipes** (Public)  
Recipes for AutoPkg  
Python 27 34
- recipes** (Public)  
Recipes for AutoPkg  
Python 214 200

Repositories

Find a repository... Type Language Sort

- autopkg** (Public)  
Automating packaging and software distribution on macOS.  
Python 1,150 192 48 7 Updated 2 days ago
- recipes** (Public)  
Recipes for AutoPkg  
Python 214 200 28 6 Updated 2 weeks ago
- homebysix-recipes** (Public)  
AutoPkg recipes all the way from Seattle, WA.  
Shell 133 Apache-2.0 111 6 4 Updated last week
- rtrouton-recipes** (Public)  
Recipes for AutoPkg  
Python 114 MIT 85 1 0 Updated 3 days ago
- dataJAR-recipes** (Public)  
Elegant and powerful Apple services for business and education.  
Python 92 Apache-2.0 69 0 2 Updated 9 hours ago

View as: Public  
You are viewing the README and pinned repositories as a public user.

People

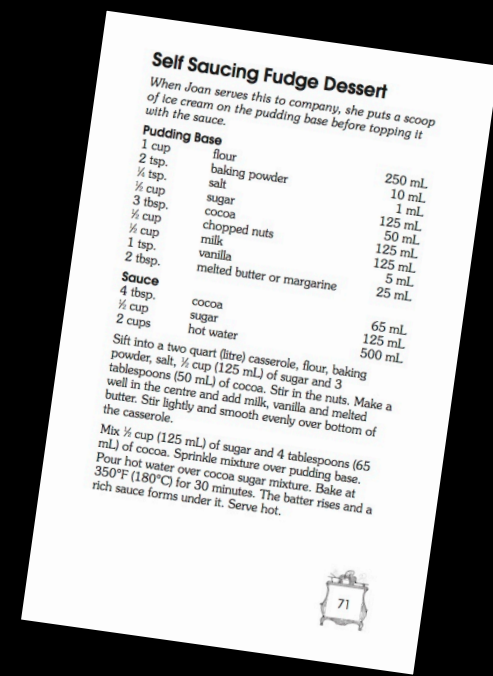
View all

Top languages

Python Shell

Most used topics Manage

autopkg python hacktoberfest apple macos



# Recipe Repositories

- AutoPkg "Core" Recipes

- ▶ <https://github.com/autopkg/recipes>

- AutoPkg GitHub "Org" Recipes

- ▶ <https://github.com/autopkg/jazzace-recipes>

- Other Public Repos

- ▶ <https://github.com/arubdesu/office-recipes>

- Private Repos

- ▶ <https://github.betterbag.com/cpe/autopkg-recipes>

# Nouns & Verbs

.download

.munki

.pkg

.jamf

.filewave

.install

# Adjectives

Parent

Child

Override

Stub

Plist

YAML

• download

Parent

• munki

• pkg

Child

Override

They're all just *recipes*

• jamf

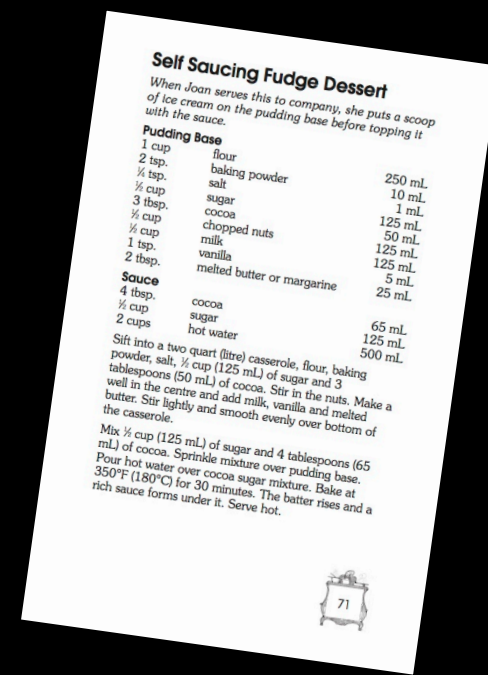
Stub

• install

Plist

• filewave

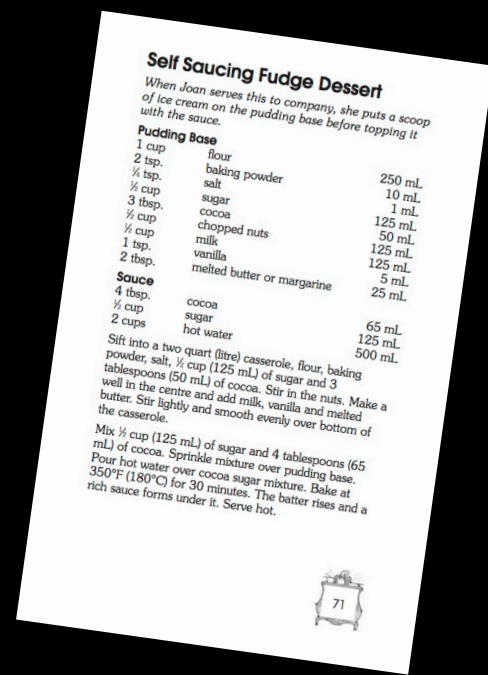
YAML



# By Relationship (Kind)

- Parent/Child
  - ▶ *Created/stored locally*
- Stub
  - ▶ *e.g., homebysix-recipes/FindAndReplace/FindAndReplace.recipe*
- Template
  - ▶ *e.g., jazzace-recipes/JamfTemplates/RecipeTemplates*





# By Function (Type)

- download

- pkg

- munki

- jamf

- Deprecated:

- ▶ *jss*

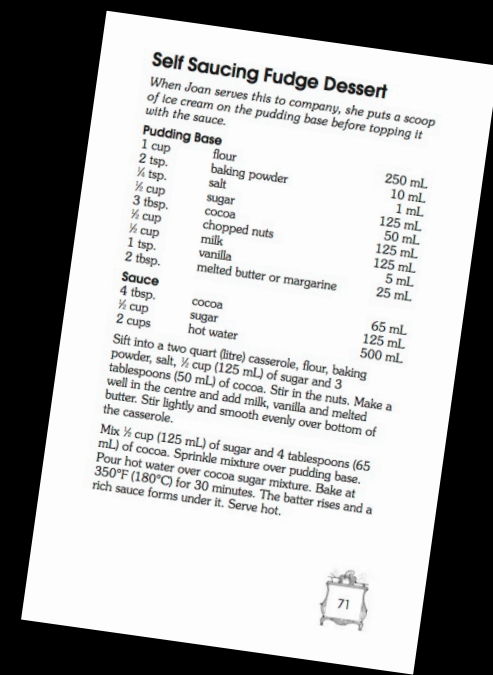
- ▶ *install*

- Others

- ▶ *verify*

- ▶ *win\**

*\* Related session: "Using AutoPkg for Windows Software 2.0"  
by James Stewart. Thursday 3:15pm in room 107.*



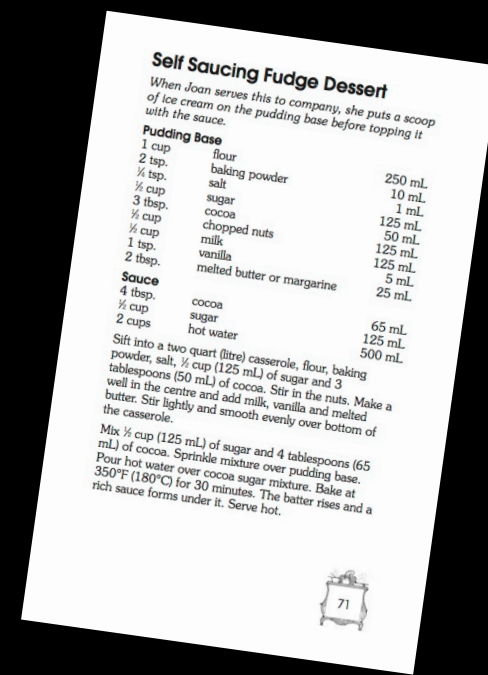
# Variables

[github.com/autopkg/autopkg/wiki/Variables](https://github.com/autopkg/autopkg/wiki/Variables)

Set via:

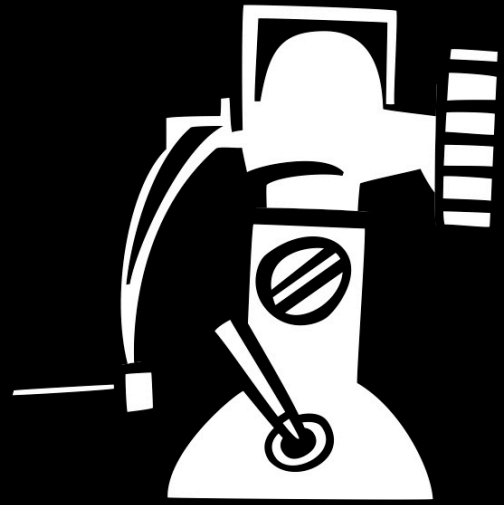
1. the Input section of the recipe (ALL\_CAPS by convention)
2. the Arguments section of a Processor call
3. output from a processor
4. AutoPkg Preferences





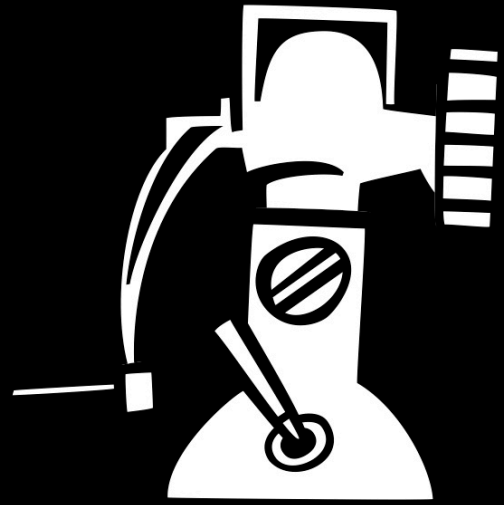
# Variables

- What “environment” do variables live in?
- Variables defined in Input
  - ▶ *Customize in your override or omit to inherit*
  - ▶ *Details: Anthony’s AutoPkg Input Variables blog post ([maclabs.jazzace.ca/2019/11/11/autopkg-input-variables.html](http://maclabs.jazzace.ca/2019/11/11/autopkg-input-variables.html))*
- Variables defined at runtime
  - ▶ *Variable substitution (%VARNAME%)*
  - ▶ *And why some processors fail (uncommon)*



# Processors (conceptually)

- Takes predictable input from previous processor[s] (or Input dict)
- Does one task well
- Produces predictable output for next processor[s]



# Processors (technically)

- Python class
- Input and/or Output
- Variables often have defaults
- Core, Custom, Shared

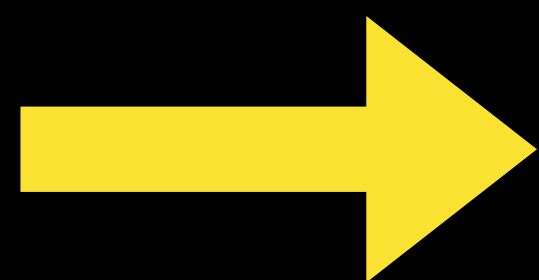


# SparkleUpdateInfoProvider

Required Input Variable (Argument)

- appcast\_url

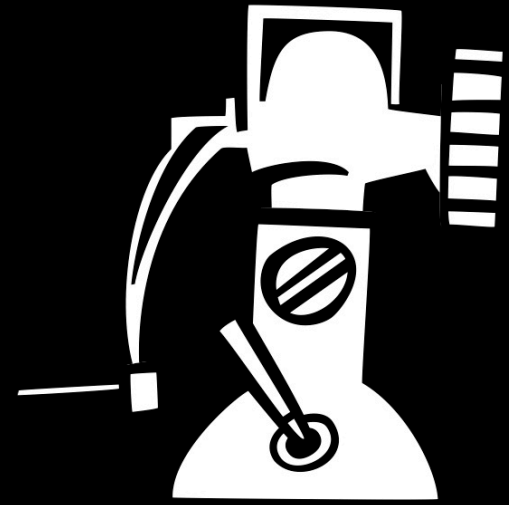
Output Variables



- url
- version
- additional\_pkginfo

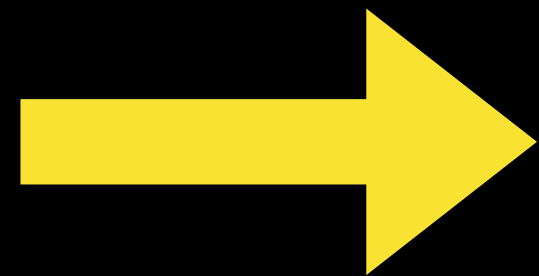
```
[...]
<key>Process</key>
<array>
  <dict>
    <key>Processor</key>
    <string>SparkleUpdateInfoProvider</string>
    <key>Arguments</key>
    <dict>
      <key>appcast_url</key>
      <string>https://ggnome.com/appcast/pano2vr/stable</string>
    </dict>
  </dict>
  <dict>
    <key>Processor</key>
    <string>URLDownloader</string>
  </dict>
</array>
[...]
```

*[https://github.com/autopkg/jazzace-recipes/blob/master/  
GardenGnome/Pano2VR.download.recipe](https://github.com/autopkg/jazzace-recipes/blob/master/GardenGnome/Pano2VR.download.recipe)*



# URLDownloader

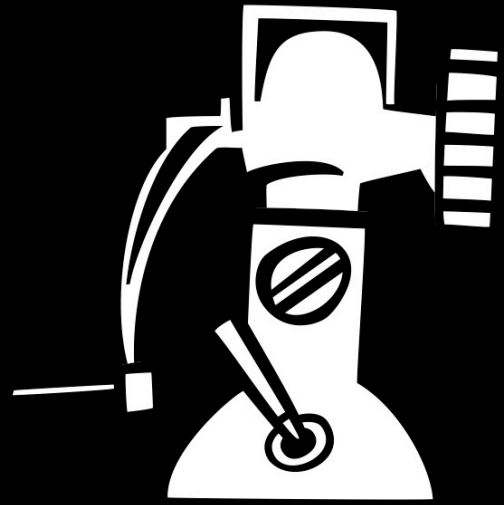
Required Input Variable (Argument)



- url

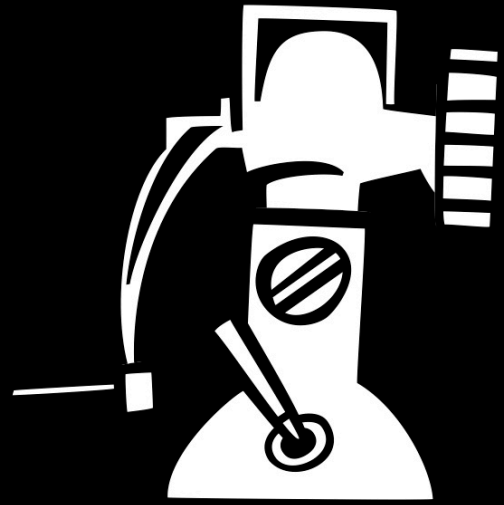
Output Variables

- pathname
- last\_modified
- url\_downloader\_summary\_result
- etag
- download\_changed



# Core Processors can...

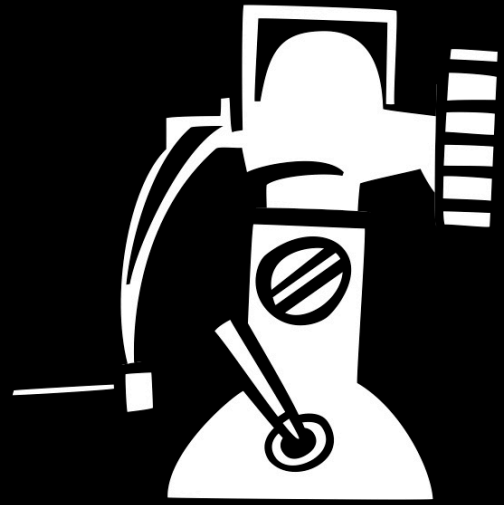
- Download things
  - ▶ *URLDownloader*
- Extract things
  - ▶ *Unarchiver, FlatPkgUnpacker, PkgPayloadUnpacker*
- Verify Code Signatures
  - ▶ *CodeSignatureVerifier*
- Move/Copy Things
  - ▶ *FileMover, Copier, PkgCopier*



# Core Processors can...

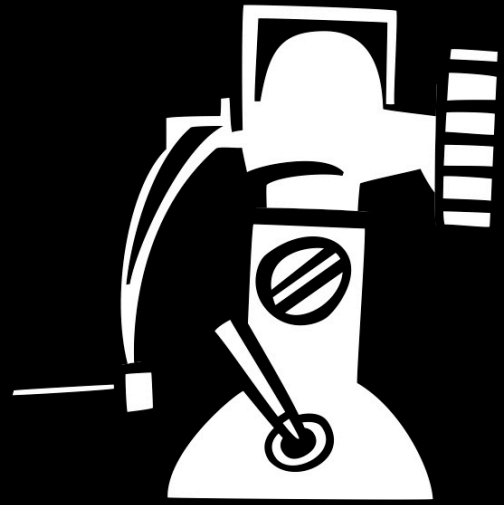
- Create things
  - ▶ *DmgCreator, PkgCreator, AppPkgCreator, FileCreator*
- Delete things
  - ▶ *PathDeleter; any processor that can overwrite*
- Parse things
  - ▶ *URLTextSearcher, SparkleUpdateInfoProvider, GitHubReleasesInfoProvider, FileFinder, AppDmgVersioner, Versioner, PlistReader*





# Core Processors can...

- Add logic or conditionals to the recipe
  - ▶ *StopProcessingIf, PackageRequired, EndOfCheckPhase, DeprecationWarning*
- Feed Munki
  - ▶ *MunkiCatalogBuilder, MunkiImporter, MunkiInfoCreator, MunkiInstallItemsCreator, MunkiOptionalReceiptEditor, MunkiPkginfoMerger, MunkiSetDefaultCatalog*
- and more!



# Core Processors

- Locally:
  - ▶ */Library/AutoPkg/autopkglib*
- Project:
  - ▶ *github.com/autopkg/autopkg/tree/master/Code/autopkglib*
- List: `autopkg list-processors`
- Highly trusted
- Documented (Wiki or `autopkg processor-info`)


NOTE: This page is automatically generated by GitHub Actions when a new release is tagged. Updates to the information on this page should be submitted as pull requests to the AutoPkg repository. Processors are located [here](#).

## Description

Downloads a URL to the specified download\_dir using curl.

## Input Variables

- url:
  - required: True
  - description: The URL to download.
- request\_headers:
  - required: False
  - description: Optional dictionary of headers to include with the download request.
- curl\_opts:
  - required: False
  - description: Optional array of options to include with the download request.
- download\_dir:
  - required: False
  - description: The directory where the file will be downloaded to. Defaults to RECIPE\_CACHE\_DIR/downloads.
- filename:
  - required: False
  - description: Filename to override the URL's tail.
- prefetch\_filename:
  - default: False
  - required: False
  - description: If true, URLDownloader attempts to determine filename from HTTP headers downloaded before the file itself. 'prefetch\_filename' overrides 'filename' option. Filename is determined from the first available source of

Table of Contents 

- [Introduction](#)
- [Getting Started](#)
  - [Windows](#)
- [FAQ](#)
- [More Resources](#)
- [Removing AutoPkg](#)
- [AutoPkg Reference](#)
  - [Preferences](#)
  - [Recipes](#)
    - [Recipe Format](#)
    - [Variables](#)
      - [Input Variables](#)
      - [Important Variable Names](#)
      - [Defining Custom Variables](#)
    - [Recipe Search Order](#)
    - [Recipe Naming Conventions](#)
    - [Recipe Overrides](#)
    - [Recipe-writing Guidelines](#)
    - [Using CodeSignatureVerification](#)
    - [Finding Recipes](#)
    - [Sharing Recipes](#)
    - [Parent Recipe Trust Information](#)
    - [Running Multiple Recipes](#)
    - [GitHub Releases with No Assets](#)
  - [Processors](#)
    - [Processor Locations](#)
    - [Pre- and postprocessors](#)
    - [Processor Summary Reporting](#)
    - [Downloading from the Internet in Custom Processors with URLGetter](#)
    - [Munki Repo Plugin Support](#)
    - [Noteworthy Shared Processors](#)
    - [Developing Custom Processors](#)
- [Processor Reference](#)
  - [AppDmgVersioner](#)
  - [AppPkgCreator](#)


NOTE: This page is automatically generated by GitHub Actions when a new release is tagged. Updates to the information on this page should be submitted as pull requests to the AutoPkg repository. Processors are located [here](#).

## Description

Downloads a URL to the specified download\_dir using curl.

## Input Variables

- url:
  - required: True
  - description: The URL to download.
- request\_headers:
  - required: False
  - description: Optional dictionary of headers to include with the download request.
- curl\_opts:
  - required: False
  - description: Optional array of options to include with the download request.
- download\_dir:
  - required: False
  - description: The directory where the file will be downloaded to. Defaults to RECIPE\_CACHE\_DIR/downloads.
- filename:
  - required: False
  - description: Filename to override the URL's tail.
- prefetch\_filename:
  - default: False
  - required: False
  - description: If true, URLDownloader attempts to determine filename from HTTP headers downloaded before the file itself. 'prefetch\_filename' overrides 'filename' option. Filename is determined from the first available source of

Table of Contents 

- [Introduction](#)
- [Getting Started](#)
  - [Windows](#)
- [FAQ](#)
- [More Resources](#)
- [Removing AutoPkg](#)
- [AutoPkg Reference](#)
  - [Preferences](#)
  - [Recipes](#)
    - [Recipe Format](#)
    - [Variables](#)
      - [Input Variables](#)
      - [Important Variable Names](#)
      - [Defining Custom Variables](#)
    - [Recipe Search Order](#)
    - [Recipe Naming Conventions](#)
    - [Recipe Overrides](#)
    - [Recipe-writing Guidelines](#)
    - [Using CodeSignatureVerification](#)
    - [Finding Recipes](#)
    - [Sharing Recipes](#)
    - [Parent Recipe Trust Information](#)
    - [Running Multiple Recipes](#)
    - [GitHub Releases with No Assets](#)
  - [Processors](#)
    - [Processor Locations](#)
    - [Pre- and postprocessors](#)
    - [Processor Summary Reporting](#)
    - [Downloading from the Internet in Custom Processors with URLGetter](#)
    - [Munki Repo Plugin Support](#)
    - [Noteworthy Shared Processors](#)
    - [Developing Custom Processors](#)
- [Processor Reference](#)
  - [AppDmgVersioner](#)
  - [AppPkgCreator](#)

# Update description of PkgRootCreator #447

Edit

<> Code ▾

Merged

 gregneagle merged 1 commit into autopkg:master from jazzace:master on Jan 21, 2019

Conversation 0

Commits 1

Checks 0

Files changed 1

+2 -1

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾

Review in codespace

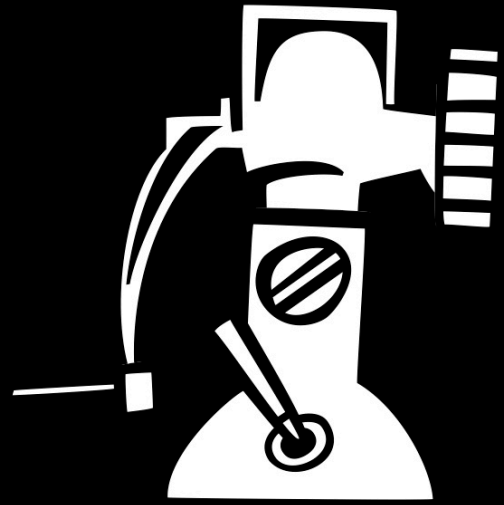
Review changes ▾

Update description of PkgRootCreator

jazzace committed on Jan 21, 2019 commit e63627984b0357596879b35bcd51cf84399ea762

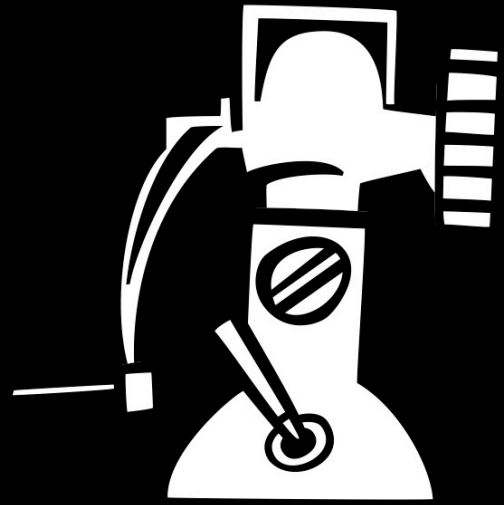
3 Code/autopkglib/PkgRootCreator.py

		@@ -29,7 +29,8 @@
29	29	
30	30	
31	31	class PkgRootCreator(Processor):
32	-	"""Creates a package root and a directory structure."""
	32	+ """Creates a package root and a directory structure.
	33	+ (Can also be used to create directory structures for other purposes.)"""
33	34	description = __doc__
34	35	input_variables = {
35	36	"pkgroot": {



# Custom Processors

- Perform tasks not covered by Core processors
- Maintained by an individual (or organization)
- Less trusted
- Kept with recipe that uses it



# Shared Processors

- Custom processor with a stub recipe
  - ▶ e.g., *com.github.homebysix.VersionSplitter/VersionSplitter*
- Custom processor that you can reference
  - ▶ e.g., *com.github.jessepeterson.munki.GrandPerspective/SourceForgeURLProvider*

# Putting it all together







# Searching for Recipes

- autopkg search  
(AutoPkgr: Search for a recipe on GitHub)
  - ▶ *Does not work reliably without adding a GitHub token*
  - ▶ *Future AutoPkg release may search internally*
- [github.com/autopkg](https://github.com/autopkg), then Search within org
  - ▶ `https://github.com/search?type=code&q=org%3Aautopkg+firefox`

```
% autopkg search Firefox | grep pkg\.recipe
MultiFirefox.pkg.recipe      ahouseini-recipes
EMCSyncplicity.pkg.recipe    bkerns-recipes
EMCSyncplicitySSO.pkg.recipe bkerns-recipes
TechSmithSnagit.pkg.recipe   bkerns-recipes
StuffItExpander.pkg.recipe    bkerns-recipes
firefoxesr.pkg.recipe         gilbertpalau-recipes
FirefoxAutoconfig.pkg.recipe gregneagle-recipes
FirefoxESRPolicies.pkg.recipe mosen-recipes
FirefoxPolicies.pkg.recipe    neilmartin83-recipes
Firefox.pkg.recipe            recipes
Firefox.nupkg.recipe          recipes
FirefoxSignedPkg.pkg.recipe   recipes
Firefox.pkg.recipe            rtrouton-recipes
MozillaThunderbird.pkg.recipe rtrouton-recipes
FirefoxPrefs.pkg.recipe       scriptingosx-recipes
Mozilla_Firefox.pkg.recipe.yaml smithjw-recipes
```

```
% autopkg search Firefox | grep pkg\.recipe
```

```
MultiFirefox.pkg.recipe  
EMCSyncplicity.pkg.recipe  
EMCSyncplicitySSO.pkg.recipe  
TechSmithSnagit.pkg.recipe  
StuffItExpander.pkg.recipe  
firefoxesr.pkg.recipe  
FirefoxAutoconfig.pkg.recipe  
FirefoxESRPolicies.pkg.recipe  
FirefoxPolicies.pkg.recipe  
→ Firefox.pkg.recipe  
Firefox.nupkg.recipe  
FirefoxSignedPkg.pkg.recipe  
→ Firefox.pkg.recipe  
MozillaThunderbird.pkg.recipe  
FirefoxPrefs.pkg.recipe  
Mozilla_Firefox.pkg.recipe.yaml
```

```
ahousseini-recipes  
bkerns-recipes  
bkerns-recipes  
bkerns-recipes  
bkerns-recipes  
gilbertpalau-recipes  
gregneagle-recipes  
mosen-recipes  
neilmartin83-recipes  
recipes  
recipes  
recipes  
rtrouton-recipes  
rtrouton-recipes  
scriptingosx-recipes  
smithjw-recipes
```



org:autopkg Firefox AND .pkg.recipe

Filter by

<> Code21

Repositories0

Issues1

Pull requests1

Discussions0

Users0

More

Languages

XML

YAML

Markdown

More languages...

Repositories

21 files (168 ms) in autopkg X

autopkg/recipes · Mozilla/Firefox.pkg.recipe

5<key>Description</key>

6<string>Downloads Firefox disk image and builds a package.

7Some useful values for RELEASE are: 'latest', 'esr-latest', 'beta-latest'.

10See the following URLs for more info:

11http://ftp.mozilla.org/pub/firefox/releases/latest/README.txt

12http://ftp.mozilla.org/pub/firefox/releases/latest-esr/README.txt

Show 7 more matches

autopkg/smithjw-recipes · Mozilla/Mozilla\_Firefox.pkg.recipe.yaml

1Description: Downloads the latest version of Firefox and renames it with tl

2Identifier: com.github.smithjw.pkg.Mozilla\_Firefox

3ParentRecipe: com.github.autopkg.download.FirefoxSignedPkg

4MinimumVersion: '2.3'

5

6Input:

7NAME: Mozilla Firefox

*Search for: org:autopkg Firefox AND .pkg.recipe*



# Dare to Compare

- Location/repo
- Broad or Narrow Focus
- Compare contents
  - ▶ Use Graham Pugh's `plistyamlplist` tool if different markup languages
  - ▶ `plutil -convert`
  - ▶ Extract the Processor names
- Maintenance (when, frequency, whom)



# Dare to Compare

## 1. **com.github.autopkg.pkg.Firefox\_EN (recipes)**

- ▶ *github.com/autopkg/recipes/blob/master/Mozilla/Firefox.pkg.recipe*
- ▶ *~/Library/AutoPkg/RecipeRepos/com.github.autopkg.recipes/Mozilla/Firefox.pkg.recipe*

## 2. **com.github.rtrouton.pkg.Firefox (rtrouton-recipes)**

- ▶ *github.com/autopkg/rtrouton-recipes/blob/master/Firefox/Firefox.pkg.recipe*
- ▶ *~/Library/AutoPkg/RecipeRepos/com.github.autopkg.rtrouton-recipes/Firefox/Firefox.pkg.recipe*

com.github.autopkg.pkg.Firefox\_EN

# **com.github.autopkg.pkg.Firefox\_EN Processors**

## ***Firefox.download***

MozillaURLProvider

URLDownloader

EndOfCheckPhase

CodeSignatureVerifier

## ***Firefox.pkg***

AppDmgVersioner

PkgRootCreator

Copier

PkgCreator



Firefox-114.0.2.p...

Back/Forward

Path

Action

Get Info

Quick Look

Installer

Search

Search

Exports

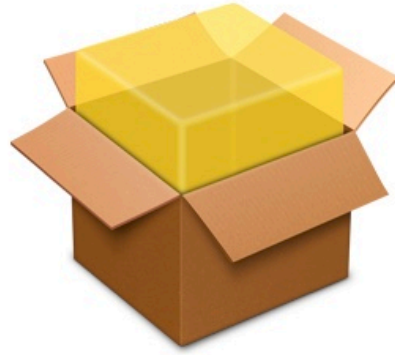
Review

Package Info

All Files

No Scripts

Receipts



Firefox-114.0.2.pkg

COMPONENT PACKAGE ?

Installs 149 items — 375.3 MB on disk

Not signed

Doesn't run any install scripts

All executable items support Apple Silicon and Intel processors

20 entitled executable items — 6 entitlements requested

Found one warning for review ?

Firefox-114.0.2.p...

Back/Forward

Path

Action

Get Info

Quick Look

Installer

Search

Search

Exports

Review


Package Info

All Files

No Scripts

Receipts

Name	Date Modified	Size	Kind
Applications	Today at 9:55 PM	375.3 MB	Folder
Firefox.app	June 19, 2023 at...	375.3 MB	Application



Name

Firefox.app

Kind

Application

Executable

Apple Silicon — 64-bit  
Intel — 64-bit

Size

375.3 MB

Modified

June 19, 2023 at 4:11 AM

Owner

root

com.github.rtrouton.pkg.Firefox

# **com.github.rtrouton.pkg.Firefox Processors**

***FirefoxSignedPkg.download***

URLTextSearcher

URLDownloader

EndOfCheckPhase

CodeSignatureVerifier

***Firefox.pkg***

FlatPkgUnpacker

FileFinder

PkgPayloadUnpacker

Versioner

PkgCopier

PathDeleter

Back/Forward

Mozilla-Firefox-1...

Path

Action

Get Info

Quick Look

Installer

Search

Search

Exports

Review

Package Info

All Files

All Scripts

Receipts



# Mozilla-Firefox-114.0.2.pkg

DEVELOPER ID SOFTWARE • DISTRIBUTION ARCHIVE ?

Installs 148 items — 375.3 MB on disk

→

DEVELOPER ID Signed by "Developer ID Installer: Mozilla Corporation (43AQ936H96)"

→

Notarized by Apple — Stapled Ticket ?

→

Runs 1 install script

→

All executable items support Apple Silicon and Intel processors

20 entitled executable items — 6 entitlements requested

→

Didn't find any issues for review ?

Back/Forward

Mozilla-Firefox-1...

Path

Action

Get Info

Quick Look

Installer

Search

Search

Exports

Review


Package Info

All Files

All Scripts

Receipts

Name	Date Modified	Size	Kind
Applications	--	375.3 MB	Folder
Firefox.app	2023-06-19	375.3 MB	Application



Name

Firefox.app

Kind

Application

Executable

Apple Silicon — 64-bit  
Intel — 64-bit

Size

375.3 MB

Modified

June 19, 2023 at 3:50 AM

Owner

root



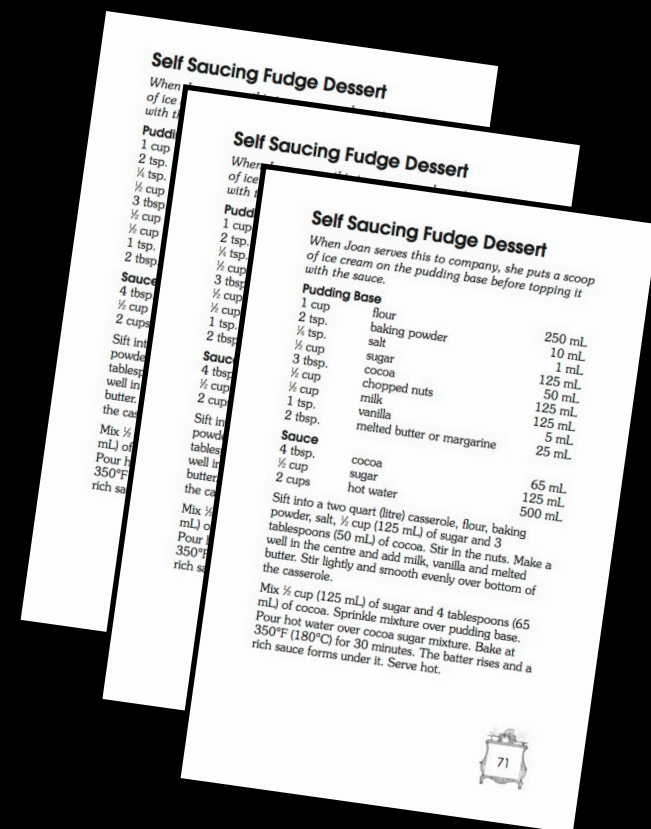
recipes

vs.



rtrouton-recipes






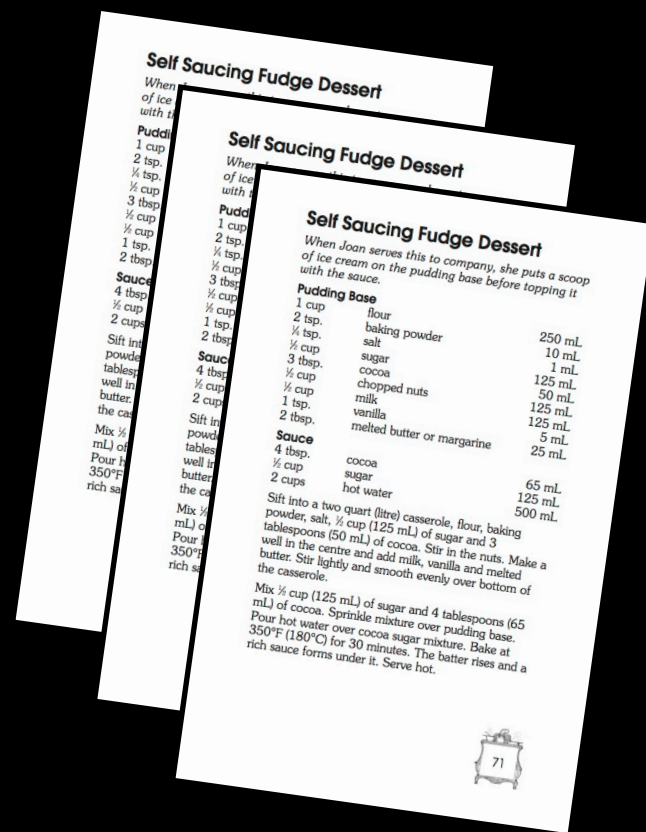
# Why Duplicates?

*In order of most to least common reason:*

- Specific need
- Existing recipe broken
- Major Version vs. Universal
- Unaware of cross-repo parent-child

# De-Duping

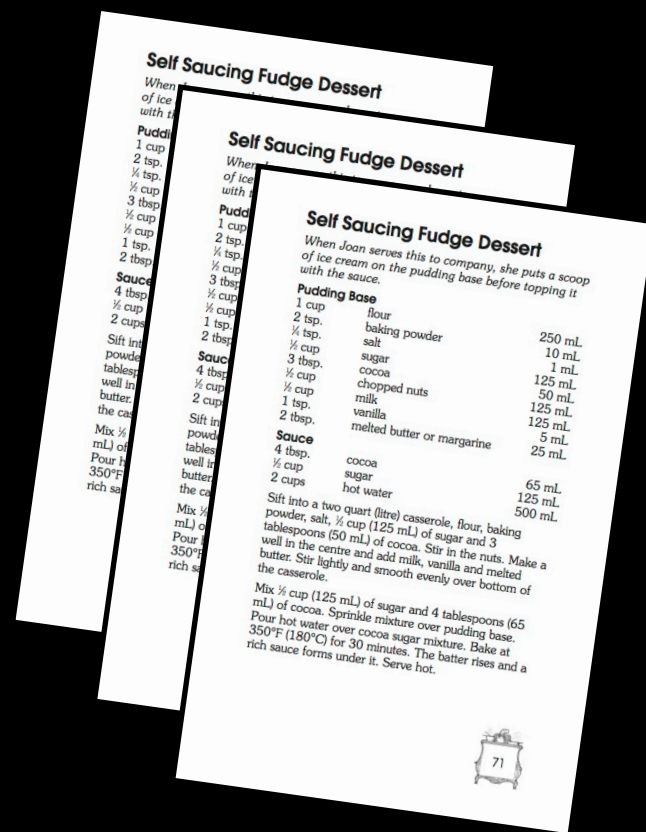
- Do different things / serve different purposes = 
- Descriptive recipe filenames to differentiate
- Maintainers (or authors) consolidate





# De-Duping Automation

- AutoPkg Maintainers might be more opinionated about duplication in the future
- An experiment in gathering and displaying duplication data:
  - ▶ <https://homebysix.github.io/autopkg-dupe-tracker/>



# Your Turn!



- Find two or more similar recipes in the AutoPkg org
- Dig into them to see how they're different
- After 10 minutes, we'll share our findings
- Optional: Discuss specific steps to resolve duplicates

# Outline

✓ AutoPkg prerequisites and basics

✓ Recipe reading and literacy

Running recipes via overrides

Specific recipe types

Creating new recipes

Testing your recipes

Sharing your recipes

Advanced topics (you tell us!)

Wrap up

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**

# Running Recipes via Overrides

# Standard Workflow

1. Search for recipe(s)

2. Add recipe repo(s)

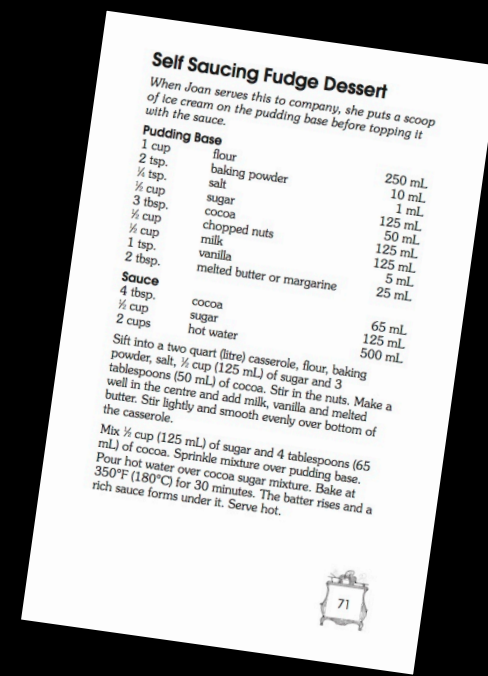
▶ *Include repos of parent recipes as needed*

3. Create override(s)

4. Run the override(s)

5. Maintain

▶ *Update repos and override trust info*



# Recipes By Relationship (Kind)

- Parent/Child

- Override

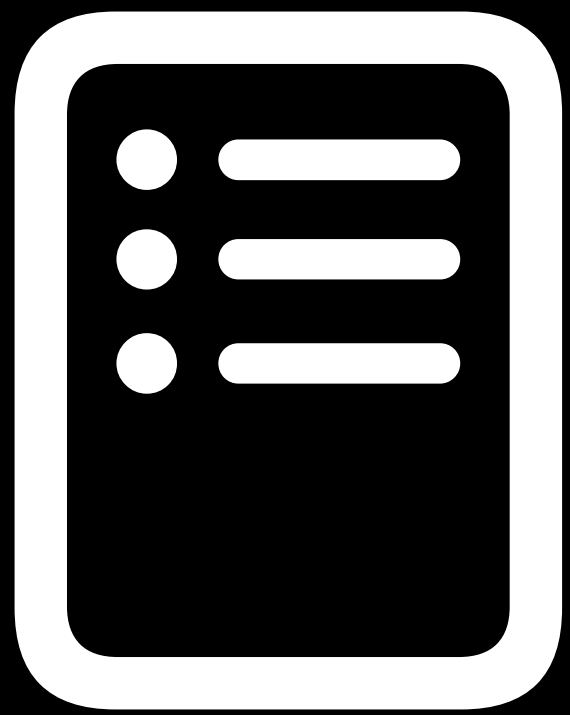
- ▶ *Created/stored locally*

- Stub

- ▶ *e.g., homebysix-recipes/FindAndReplace/FindAndReplace.recipe*

- Template

- ▶ *e.g., jazzace-recipes/JamfTemplates/RecipeTemplates*



# Why Overrides?

- Customize input variables
- Maintain trust information



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Identifier</key>
  <string>local.pkg.nothing</string>
  <key>Input</key>
  <dict>
    <key>NAME</key>
    <string>nothing</string>
  </dict>
  <key>ParentRecipe</key>
  <string>com.github.yourgithubusername.pkg.nothing</string>
  <key>ParentRecipeTrustInfo</key>
  <dict>
    <key>non_core_processors</key>
    <dict/>
    <key>parent_recipes</key>
    <dict>
      <key>com.github.yourgithubusername.download.nothing</key>
      <dict>
        <key>path</key>
        <string>~/Library/AutoPkg/Recipes/nothing.download.recipe</string>
        <key>sha256_hash</key>
        <string>fda53c4c7dbb1550aa9a888af14537bd42f16d73bb48da685b4da70fcc2e8500</string>
      </dict>
      <key>com.github.yourgithubusername.pkg.nothing</key>
      <dict>
        <key>path</key>
        <string>~/Library/AutoPkg/Recipes/nothing.pkg.recipe</string>
        <key>sha256_hash</key>
        <string>463565c8aaf8c02465b611c589d3cb83a602c288b184f3e0b49d8d435bb10406</string>
      </dict>
    </dict>
  </dict>
</dict>
</plist>
```

Identifier (typically starts with local.)

Input

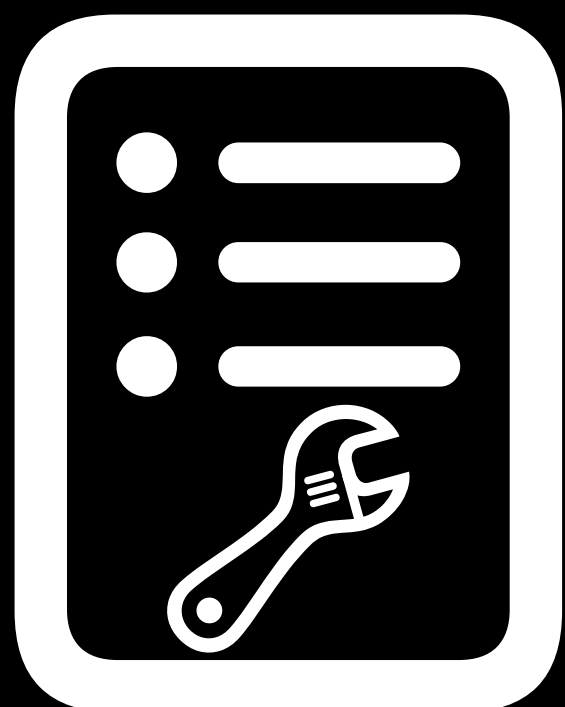
ParentRecipe

non\_core\_processors

ParentRecipeTrustInfo

parent\_recipes

No Process Array



# Variables in Overrides

- Example: Intel vs Apple Silicon Architecture

## GitHubDesktop.download.recipe (parent of parent)

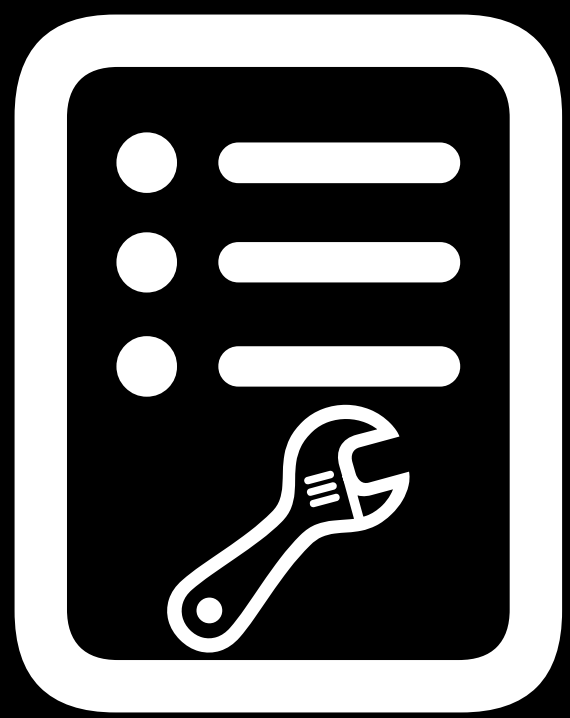
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...>
<plist version="1.0">
<dict>
  <key>Description</key>
  <string>Downloads the latest version of GitHub Desktop.
```

Valid BUILD arguments are:

- **"darwin": Downloads GitHub Desktop for Intel Macs** ←
  - "darwin-arm64": Downloads GitHub Desktop for Apple Silicon Macs
- ```
  </string>
  <key>Identifier</key>
  <string>com.github.homebysix.download.GitHubDesktop</string>
  <key>Input</key>
  <dict>
    <key>BUILD</key>
    <string>darwin</string> ←
    <key>NAME</key>
    <string>GitHub Desktop</string>
  </dict>
```

## GitHubDesktop.munki.recipe (override)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...>
<plist version="1.0">
<dict>
  <key>Identifier</key>
  <string>local.munki.GitHubDesktop</string>
  <key>Input</key>
  <dict>
    <key>BUILD</key>
    <string>darwin</string> ←
    <key>MUNKI_REPO_SUBDIR</key>
    <string>apps/GitHub</string>
    <key>NAME</key>
    <string>GitHub Desktop</string>
```



# Variables in Overrides

- Can make/run multiple overrides for same recipe

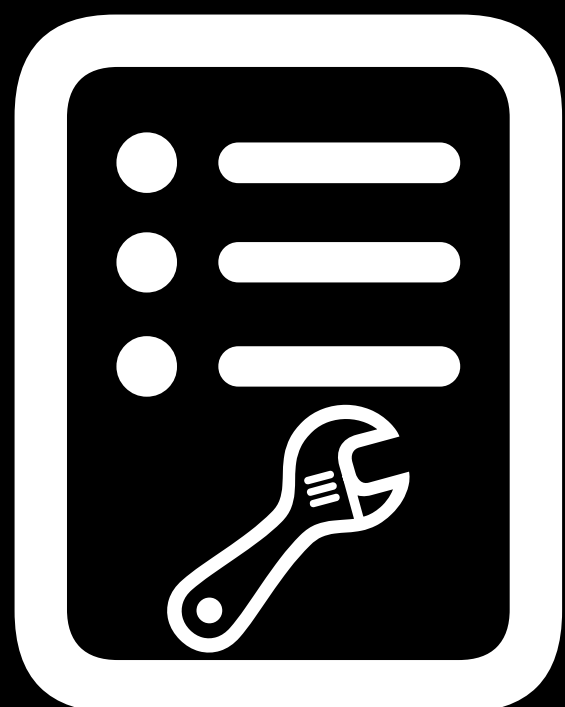
► *Example: Non-Universal: One Intel, one Apple Silicon*

## GitHubDesktop\_x86\_64.munki.recipe (override)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...>
<plist version="1.0">
<dict>
  <key>Identifier</key>
  <string>local.munki.GitHubDesktop_x84_64</string>
  <key>Input</key>
  <dict>
    <key>BUILD</key>
    <string>darwin</string>
```

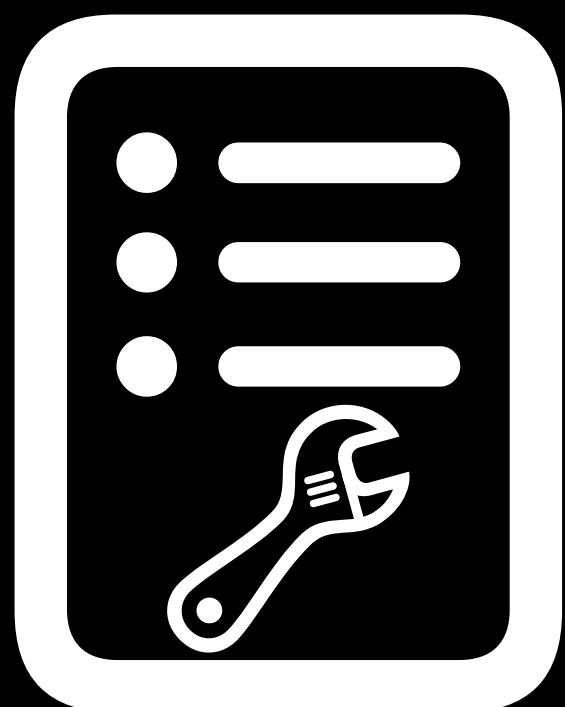
## GitHubDesktop\_arm64.munki.recipe (override)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...>
<plist version="1.0">
<dict>
  <key>Identifier</key>
  <string>local.munki.GitHubDesktop_arm64</string>
  <key>Input</key>
  <dict>
    <key>BUILD</key>
    <string>darwin-arm64</string>
```



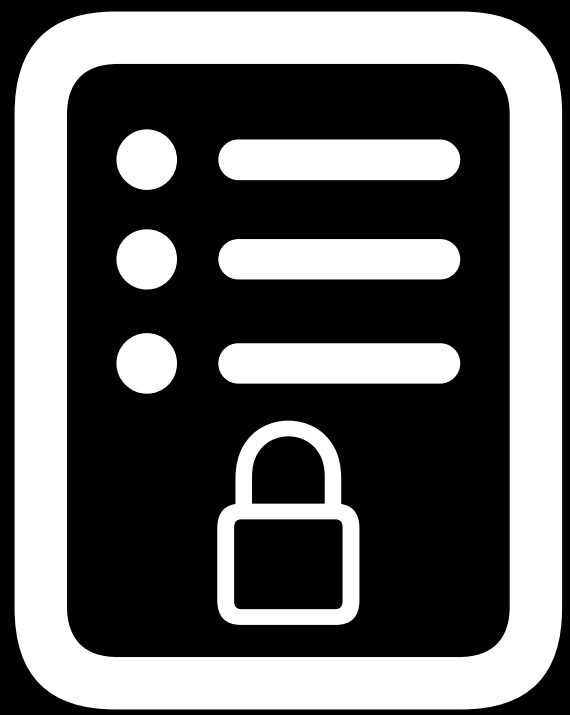
# Variables in Overrides

- Can make/run multiple overrides for same recipe
  - ▶ *Example: Non-Universal: One Intel, one Apple Silicon*
  - ▶ *Example: Firefox, FirefoxESR*
- Can use `--name` or `-n` to specify name of override
  - ▶ *Ensures unique local identifier*
  - ▶ *AutoPkgr prompts for this automatically*



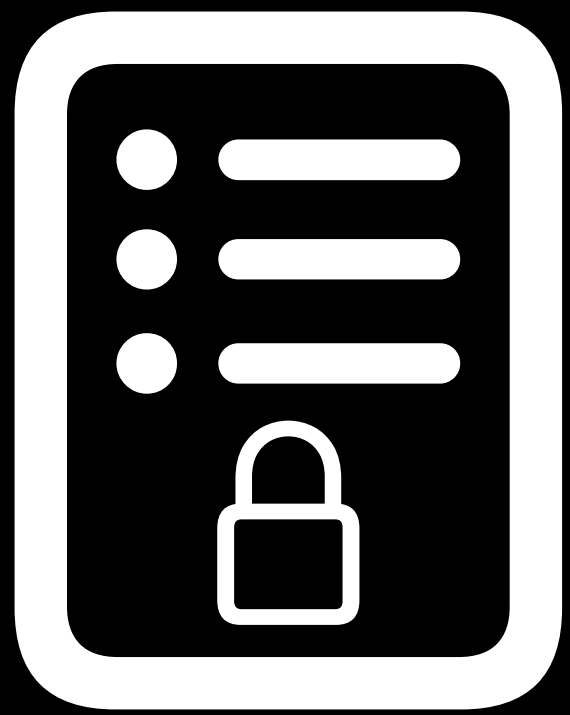
# Variables in Overrides

- When you might NOT want to override
  - ▶ *Example: Sparkle feed, download URLs*
- Munki pkginfo: all or nothing
  - ▶ *AutoPkg does per-variable substitution, and pkginfo is considered a single variable*
  - ▶ *Example: supported\_architectures*
  - ▶ *Can build flexibility into pkginfos by making one or more of their contents a separate variable*
  - ▶ *MunkiPkgInfoMerger can add additional keys/values*



# Trust Info in Overrides

- Allows you to audit changes
- Pins your process to a known state
- Canary indicator for parent updates

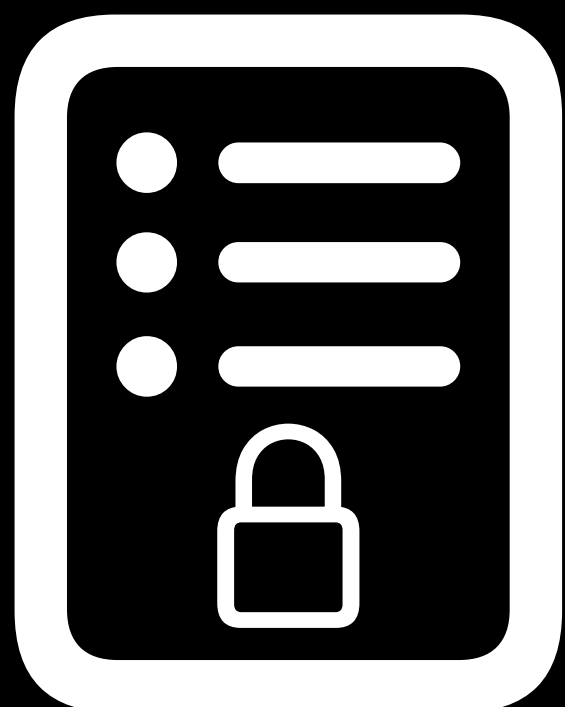


# Trust Info in Overrides

- `make-override` adds trust info
- `update-trust-info` updates trust info
- To enable strict trust verification:

```
defaults write com.github.autopkg FAIL_RECIPES_WITHOUT_TRUST_INFO -bool YES
```

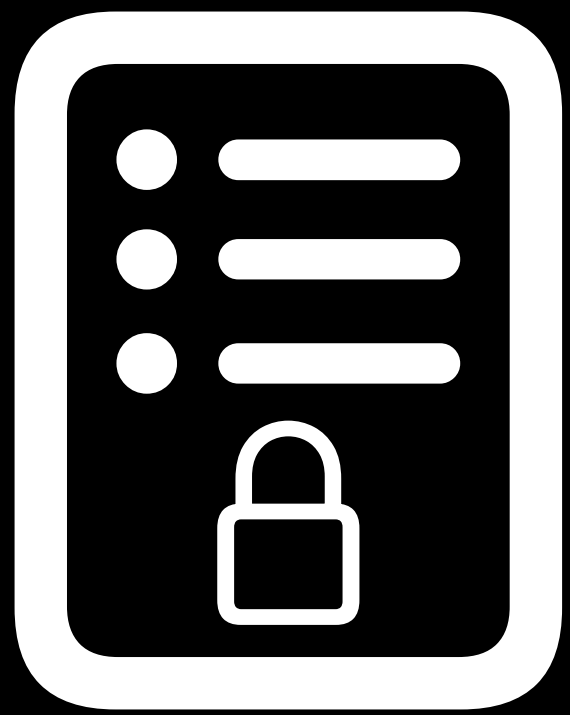




# Maintaining Trust Info

- Method A: Automatically Repo-Update
  - ▶ Run *repo-update* regularly, then respond reactively to trust failures. AutoPkgr provides a checkbox for this method.
- Method B: Manually Repo-Update
  - ▶ Run *repo-update* only when recipes fail or require troubleshooting. (Might miss out on some benefits.)

Either way: **You** are responsible for auditing changes between `repo-update` and `update-trust-info`.



# Maintaining Trust Info

- `autopkg run -vv`
- `autopkg verify-trust-info -vv`
- AutoPkgr (minimal)

# Outline

✓ AutoPkg prerequisites and basics

✓ Recipe reading and literacy

★ Running recipes via overrides – *recipe lists and scheduled runs*

Specific recipe types

Creating new recipes

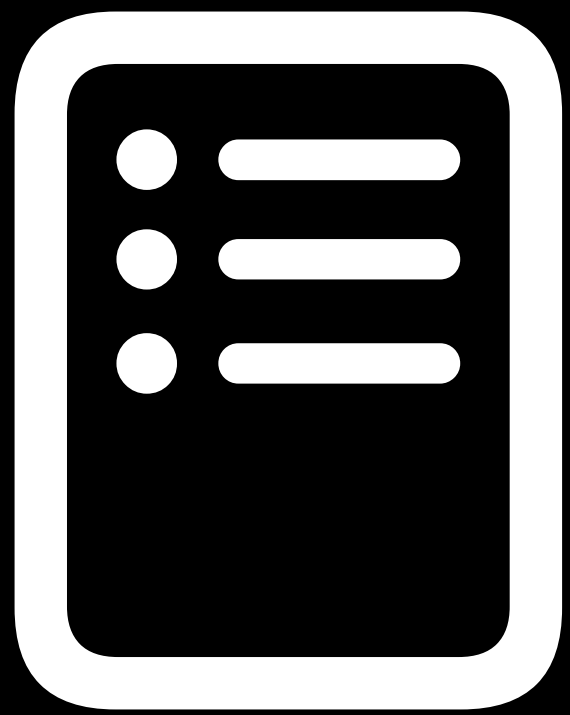
Testing your recipes

Sharing your recipes

Advanced topics (you tell us!)

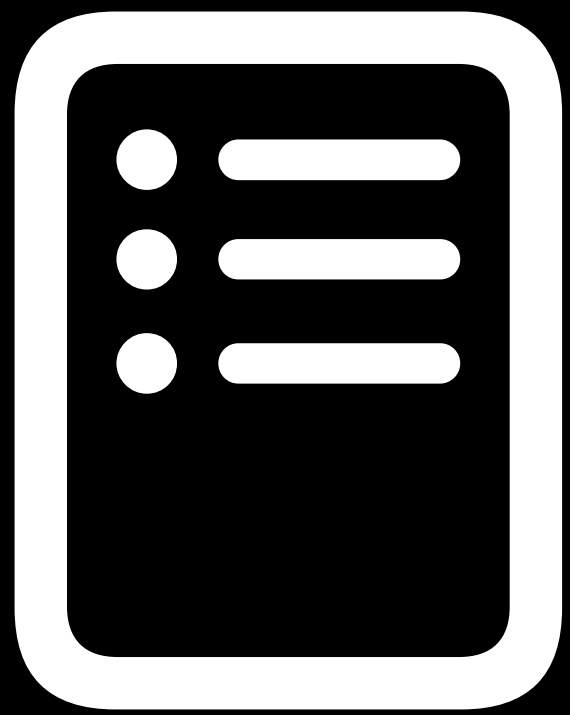
Wrap up

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**



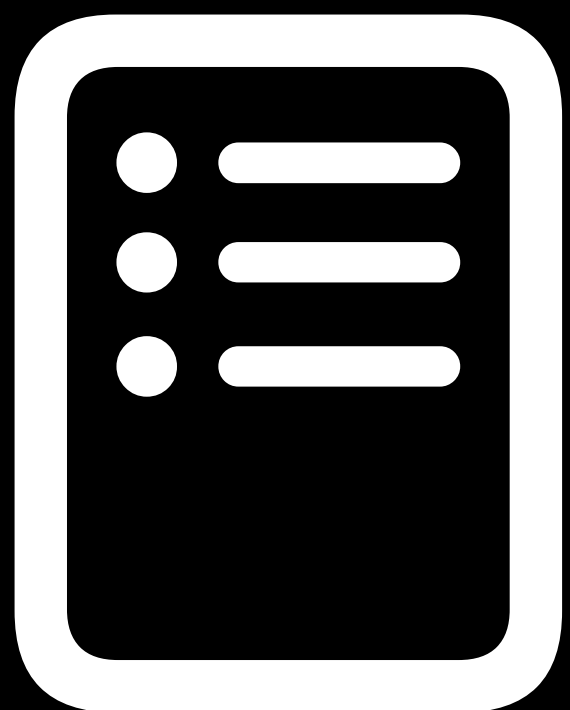
# Recipe Lists

- Collections of recipes that you run together
  - ▶ *Example: Your Munki recipes might be on a different schedule than your Jamf recipes.*
  - ▶ *Or you might have multiple Jamf servers running on different schedules.*
  - ▶ *Or you might have some recipes that require a VPN connection to do their downloading and you want to manually kick those off.*



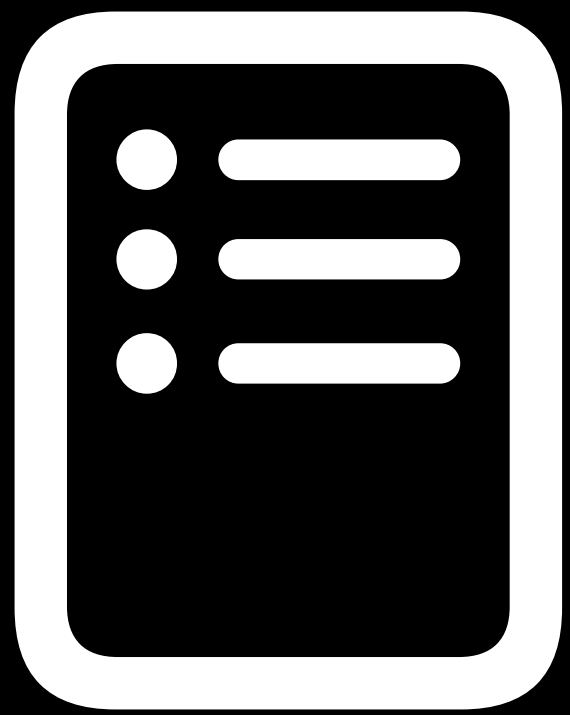
# Recipe Lists

- Use `--recipe-list` or `-l` flag
- Can be text or plist format
  - ▶ *Benefit of plist: environment variables and pre/post processors*
- Can be recipe names or full identifiers
  - ▶ *Full identifiers are never a bad idea, to avoid ambiguity*
- AutoPkgr supports multiple recipe lists



# Recipe Lists

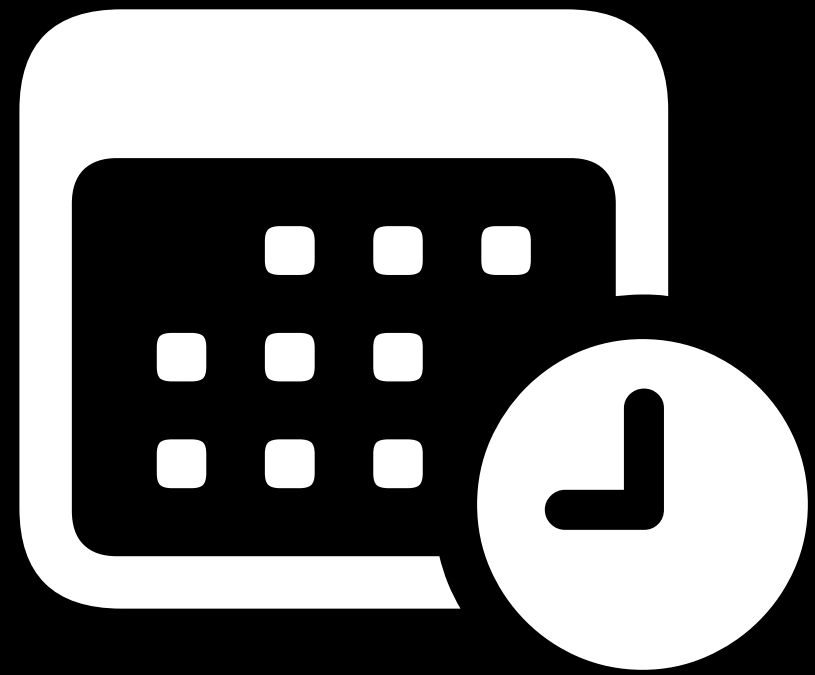
- Beware: Multiple simultaneous AutoPkg runs can fail when competing for the same resources/services
  - ▶ *Example: autopkgserver should only be called upon to package one thing at a time*
- Beware: MakeCatalogs.munki (or override) should be the final recipe in lists containing Munki recipes



# Recipe Lists

- `verify-trust-info` verb supports lists
- `update-trust-info` verb does NOT support lists (by design)
  - ▶ *Does support wildcards, but use discretion*
- Tip from Anthony:
  - ▶ *Verify recipe list, then follow up with `-vv` on failed recipes*





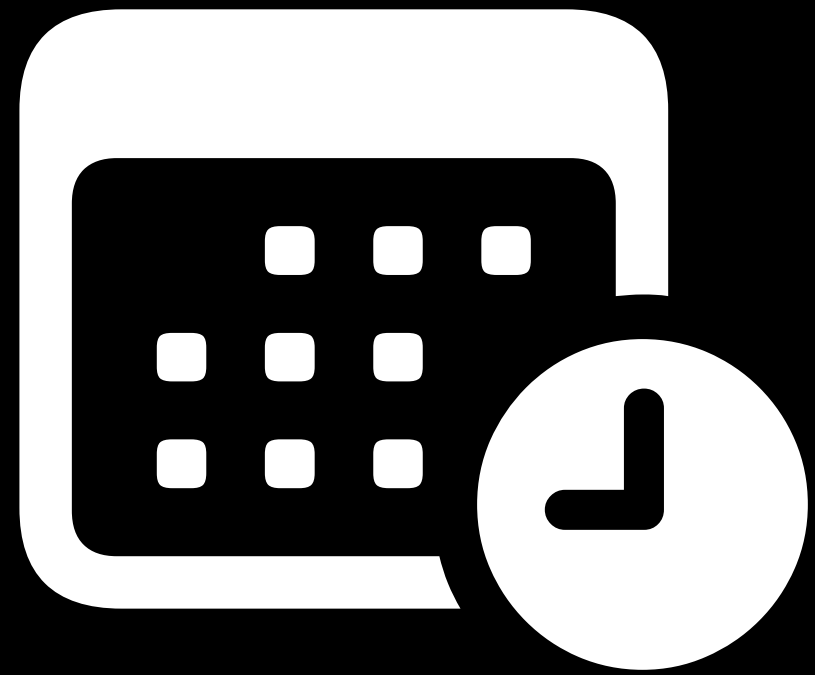
# Scheduling AutoPkg

- AutoPkgr

▶ <https://github.com/lindegrouper/autopkg/wiki/Scheduling-and-Notifications>

The screenshot shows the AutoPkgr application window with the 'Schedule' tab selected. The window has a title bar with standard macOS window controls (red, yellow, green buttons) and the title 'AutoPkgr'. Below the title bar is a navigation bar with five tabs: 'Install', 'Repos & Recipes', 'Schedule' (highlighted in blue), 'Notifications', and 'Folders & Integration'. The main content area of the 'Schedule' tab contains the following settings:

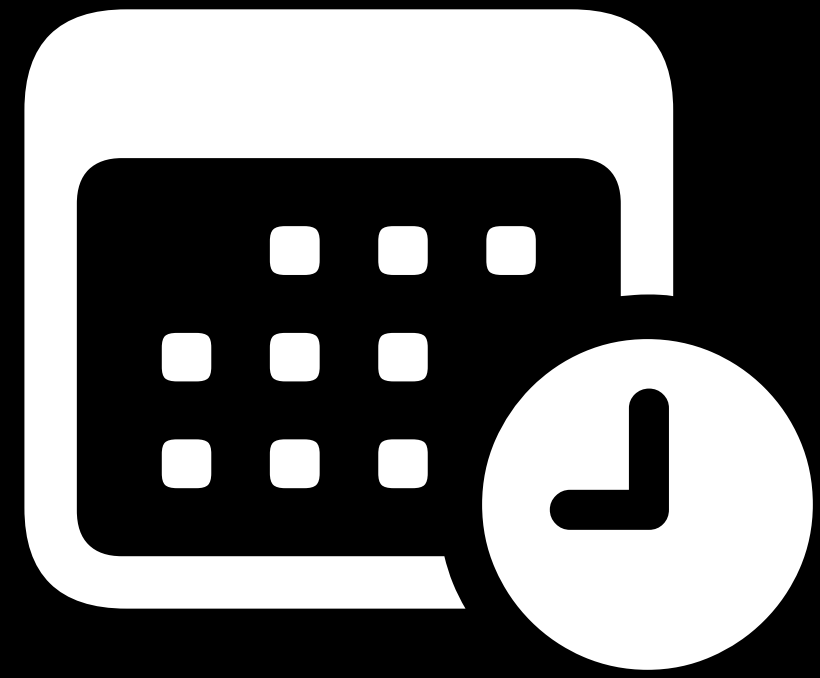
- ☐ Enable scheduled AutoPkg runs
  - ☒ Every  hours
  - ☐ Daily at
  - ☐ Weekly on   at
- ☐ Update all repos before each AutoPkg run



# Scheduling AutoPkg

- LaunchAgent and runner script

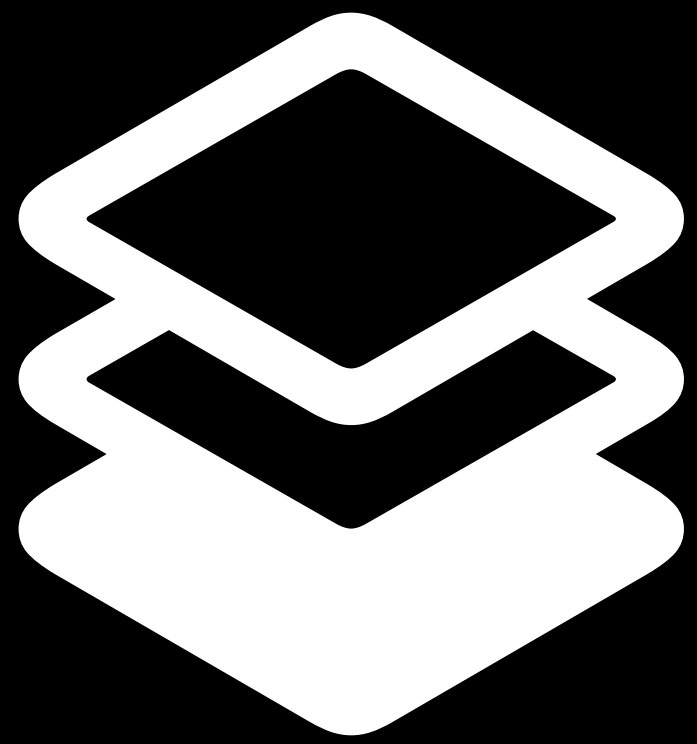
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.betterbag.cpe.autopkgrunner</string>
  <key>ProgramArguments</key>
  <array>
    <string>/Users/autopkg-runner/Developer/cpe-autopkg/autopkg-runner.sh</string>
  </array>
  <key>RunAtLoad</key>
  <true/>
  <key>StandardErrorPath</key>
  <string>/private/var/log/runner.log</string>
  <key>StandardOutPath</key>
  <string>/private/var/log/runner.log</string>
  <key>StartInterval</key>
  <integer>14400</integer>
  <key>UserName</key>
  <string>autopkg-runner</string>
</dict>
</plist>
```



# Scheduling AutoPkg

- Hosted CI/CD
  - ▶ <https://engineering.gusto.com/running-autopkg-in-github-actions/>

# Specific Recipe Types



# Specific Recipe Types

**<NAME>** . **<TYPE>** . recipe [ .yaml ]

GoogleChromePkg.download.recipe

GoogleChrome.pkg.recipe

GoogleChrome.munki.recipe

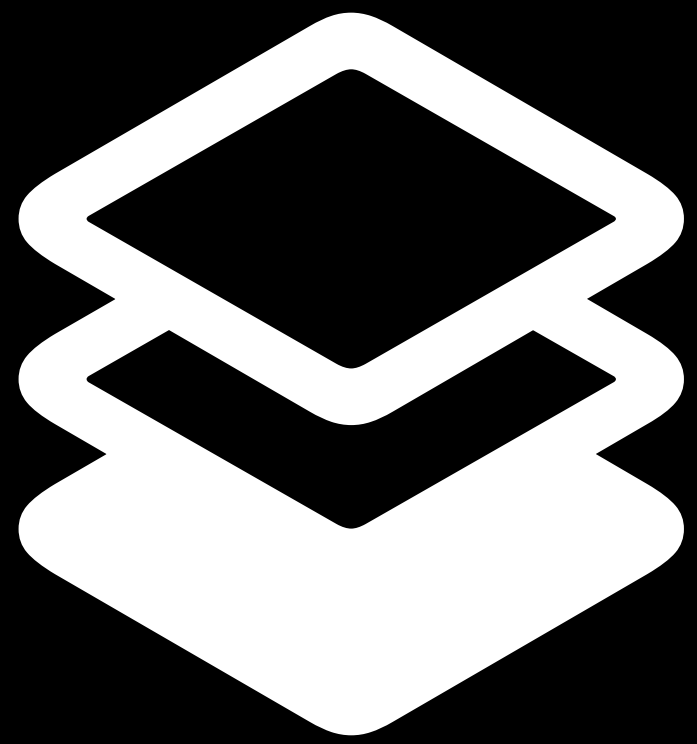
GoogleChrome.ds.recipe

Google Chrome.jss.recipe

GoogleChromeEnterprise.jss.recipe

GoogleChrome.jamf-upload.recipe.yaml

Google\_Chrome.jamf.recipe.yaml



# Specific Recipe Types

- Let's look together at some recipe type conventions
- Which do you use? Which are you curious about?
  - Download
  - Verify
  - Pkg
  - Munki
  - JSS
  - Jamf
  - DeployStudio
  - BigFix
  - Filewave
  - LANRev/Absolute Manage
  - Intune/SCCM/MEM
  - Windows
  - *Others?*

# Outline

- ✓ AutoPkg prerequisites and basics
- ✓ Recipe reading and literacy
- ✓ Running recipes via overrides
- ✓ Specific recipe types

Creating new recipes

Testing your recipes

Sharing your recipes

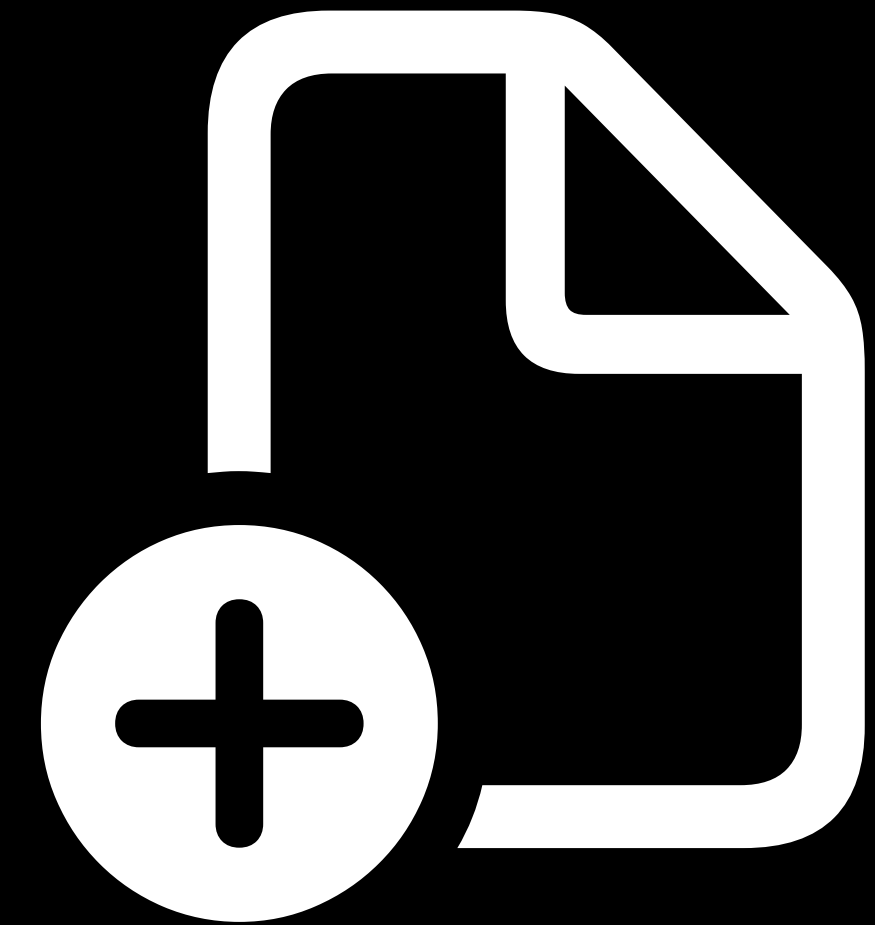
Advanced topics (you tell us!)

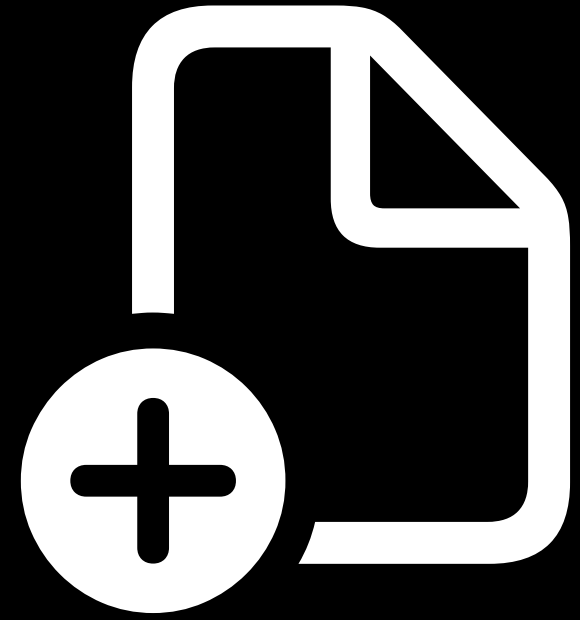
Wrap up

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**



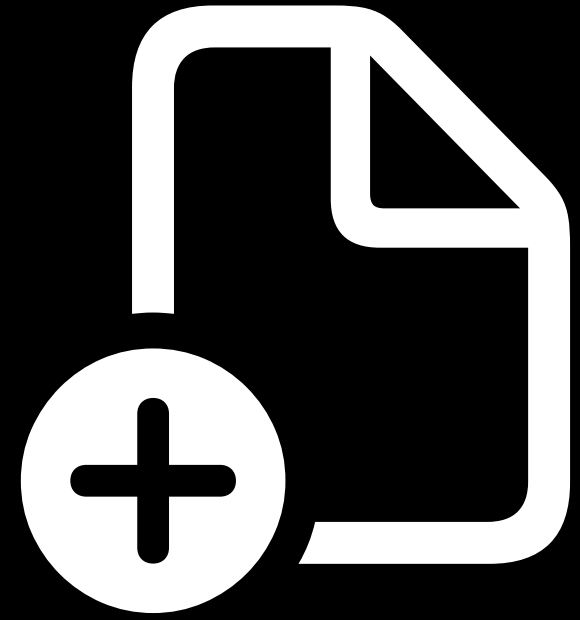
# Creating New Recipes





# First: Search Thoroughly

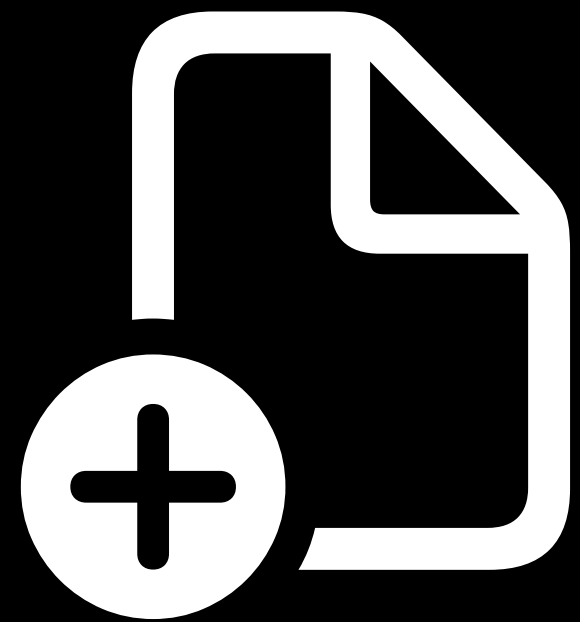
- AutoPkg search verb
  - ▶ *As mentioned before, not 100% reliable as of now*
- GitHub web search
  - ▶ *<https://github.com/search?type=code&q=org%3Aautopkg+thunderbird>*



# Creating a Recipe

- ~~Steal~~ Adapt an existing recipe
- Recipe Robot
- Write it from scratch





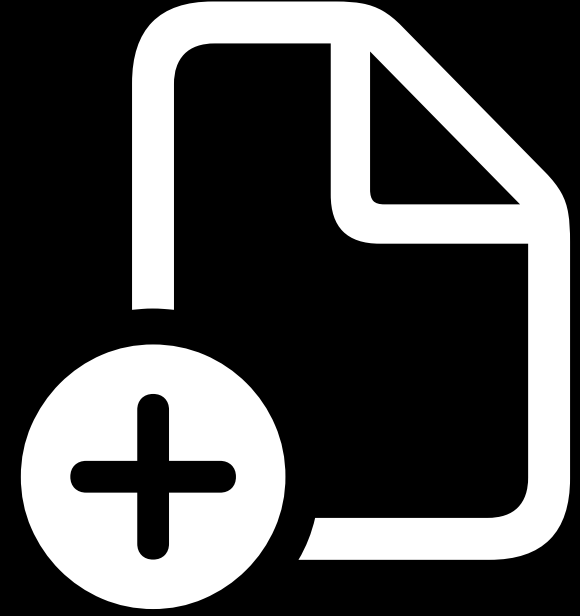
# Creating a Recipe

- ~~Steal~~ Adapt an existing recipe
- Recipe Robot
- Write it from scratch



`autopkg new-recipe recipeName.function.recipe`

- Leverage an existing parent recipe if you can

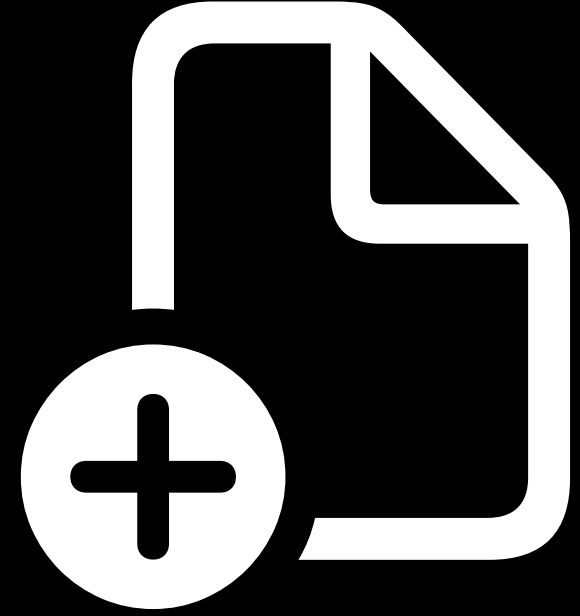


# Creating a Recipe

- Start by determining how app is downloaded
  - ▶ *If app isn't publicly downloadable, creating recipes isn't impossible but result will be less automated*
  - ▶ *PackageRequired may be an option*

```
...
<key>Process</key>
<array>
  <dict>
    <key>Processor</key>
    <string>PackageRequired</string>
  </dict>
  <dict>
    <key>Processor</key>
    <string>AppPkgCreator</string>
    <key>Arguments</key>
    <dict>
      <key>app_path</key>
      <string>%PKG%/*.app</string>
      <key>force_pkg_build</key>
      <true/>
    </dict>
  </dict>
</array>
</dict>
</plist>
```

*[maclabs.jazzace.ca/2018/11/17/generic-packaging-recipes-for-autopkg.html](http://maclabs.jazzace.ca/2018/11/17/generic-packaging-recipes-for-autopkg.html)*



# Creating a Recipe

- Many recipes share same DNA
  - ▶ *Static download URL, zipped app, unsigned:*
    - URLDownloader, EndOfCheckPhase
  - ▶ *Sparkle feed, dmg app, signed*
    - SparkleUpdateInfoProvider, URLDownloader, EndOfCheckPhase, CodeSignatureVerifier

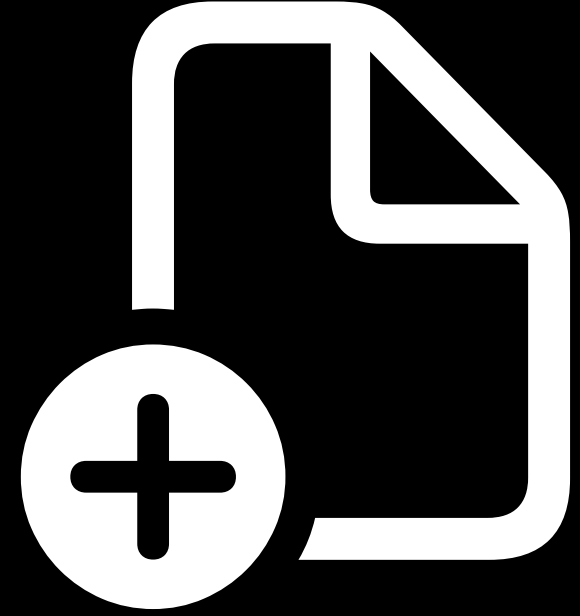




# Recipe Robot

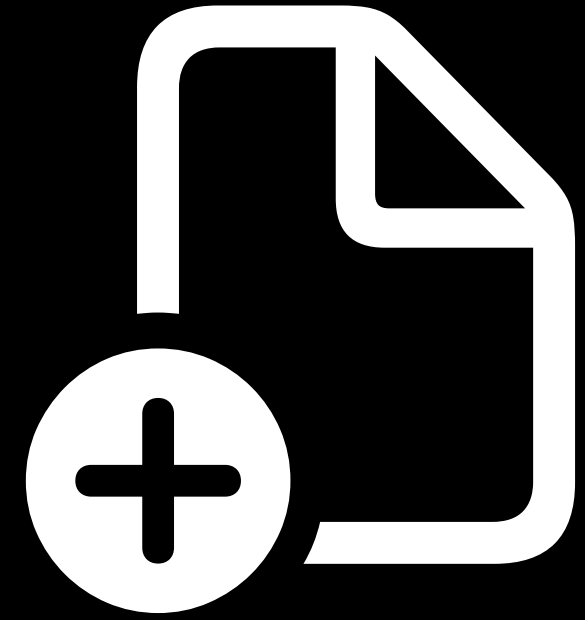
<https://github.com/homebysix/recipe-robot>

- Provide an input to get started
  - ▶ *App, download URL, GitHub repo, etc.*
- Works best with simple, conventional apps
  - ▶ *Public download, dmg or zip format, installs to /Applications*
- Can only create entire recipe “chains”
  - ▶ *But can create a chain and splice into existing parents*



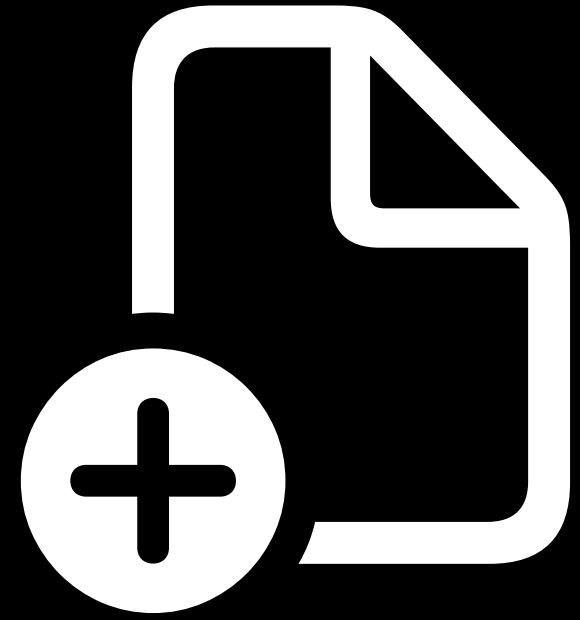
# New-Recipe Verb

- Better than starting from scratch
- Copy/paste processor blocks from existing recipes as needed



# Common Download Recipe Workflow

1. Determine the URI of the item we want to download
2. Download a copy of the item
3. EndOfCheckPhase
4. Check the code signature of the downloaded item
  - Sometimes: Collect version information



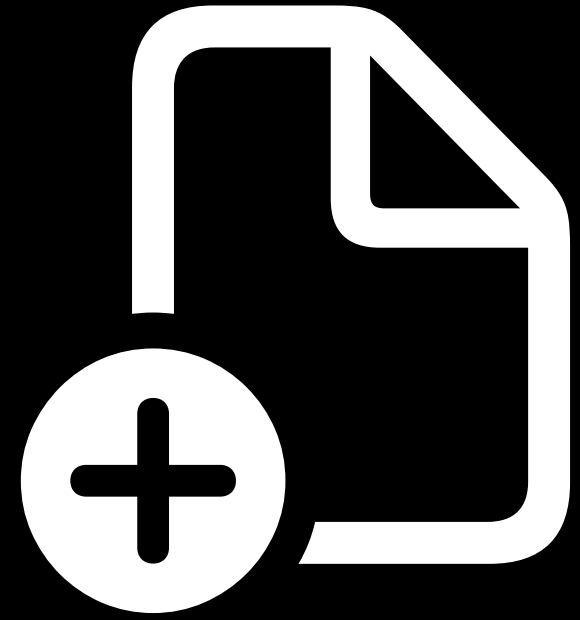
# Retrieve Code Signature

- Application Bundle (.app)
  - ▶ `codesign --display -r- --deep /path/to/the.app`
  - ▶ *If .zip, unarchive first (Unarchiver processor in recipe)*
  - ▶ *If .dmg, mount disk image first*
- Package Installer (.pkg)
  - ▶ `pkgutil --check-signature /path/to/the.pkg`

<https://github.com/autopkg/autopkg/wiki/Using-CodeSignatureVerification/>

```
<dict>
  <key>Processor</key>
  <string>CodeSignatureVerifier</string>
  <key>Arguments</key>
  <dict>
    <key>comment</key>
    <string>Use wildcard matching for the app so it will also
match FirefoxDeveloperEdition.app and FirefoxNightly.app</string>
    <key>input_path</key>
    <string>%pathname%/Firefox*.app</string>
    <key>requirement</key>
    <string>anchor apple generic and certificate
leaf[field.1.2.840.113635.100.6.1.9] /* exists */ or anchor apple
generic and certificate 1[field.1.2.840.113635.100.6.2.6] /*
exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /*
exists */ and certificate leaf[subject.OU] = "43AQ936H96"</string>
  </dict>
</dict>
```

```
<dict>
  <key>Arguments</key>
  <dict>
    <key>expected_authority_names</key>
    <array>
      <string>Developer ID Installer: Mozilla Corporation
(43AQ936H96)</string>
      <string>Developer ID Certification Authority</string>
      <string>Apple Root CA</string>
    </array>
    <key>input_path</key>
    <string>%pathname%</string>
  </dict>
  <key>Processor</key>
  <string>CodeSignatureVerifier</string>
</dict>
```



# URLTextSearcher

## Input Variables

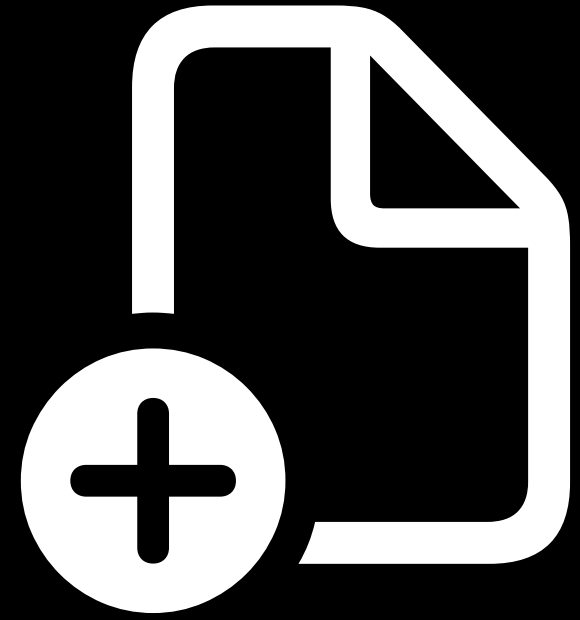
- `re_pattern*`
- `url*`
- `result_output_var_name`
- `request_headers`
- `curl_opts`
- `re_flags`

## Output Variable

- `result_output_var_name`

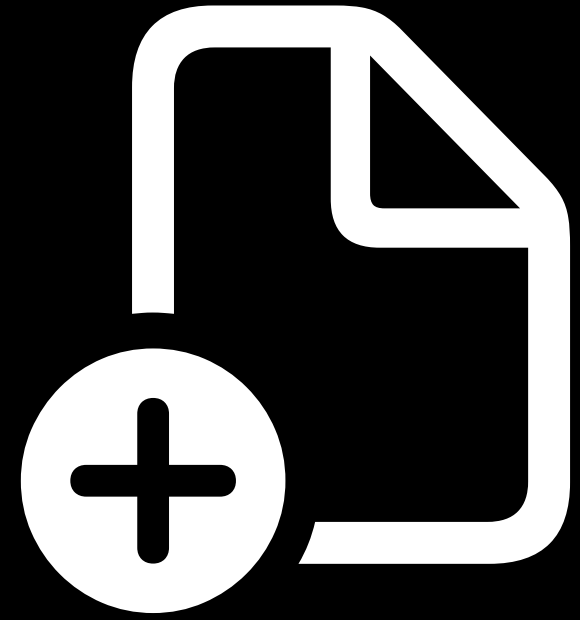
`* required`





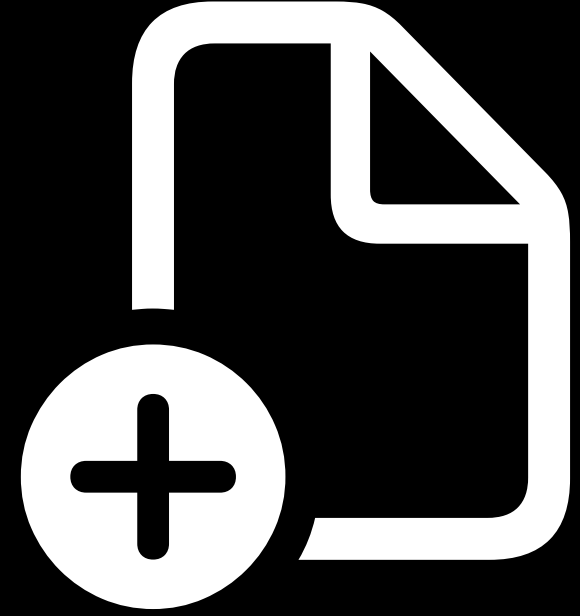
# Start the Search

- Save the web page
  - ▶ *Safari: Save As... > Page Source*
  - ▶ *Chrome: Save Page As... > Webpage, HTML Only*
  - ▶ *Firefox: Save Page As... > Web Page, HTML only*
  - ▶ *All browsers: Web inspector*
  - ▶ `curl -sL <url>`
- Open in your text editor of choice



# RegEx

- Test in your text editor
  - ▶ *Borrow techniques from other recipes*
  - ▶ *Leverage text editor features (e.g., BBEdit live regex)*
  - ▶ *William Smith: An Introduction to regex*  
*(MacAdmins Campfire Sessions 2020)*
    - GitHub Repo: [github.com/talkingmoose/introduction-to-regex](https://github.com/talkingmoose/introduction-to-regex)
    - Session Video: [youtu.be/Wc8Kpw0nEww](https://youtu.be/Wc8Kpw0nEww)
- Other options: Patterns.app, regex101.com



# RegEx

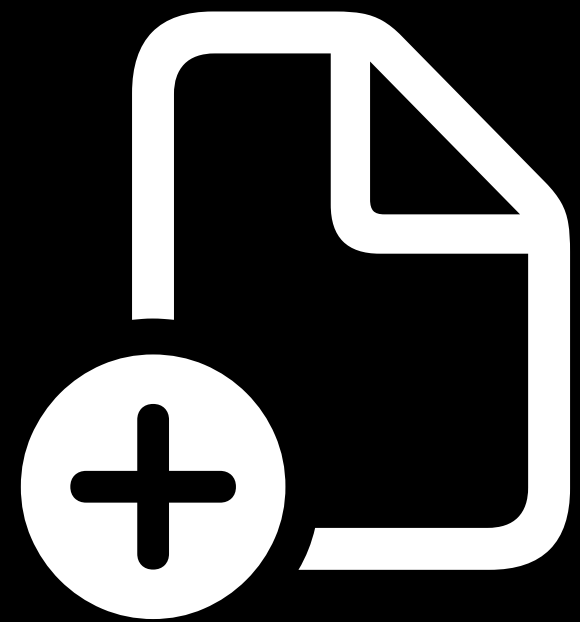
- Test in your recipe
  - ▶ *Plist: encode < and > (&lt; &gt;)*
  - ▶ *YAML: single-quote strings with colon space (: ) inside or that begin with " or %*





[macmule.com/2021/06/14/macdevopsyvr-2021-digging-for-download-urls-for-autopkg-recipes/](https://macmule.com/2021/06/14/macdevopsyvr-2021-digging-for-download-urls-for-autopkg-recipes/)





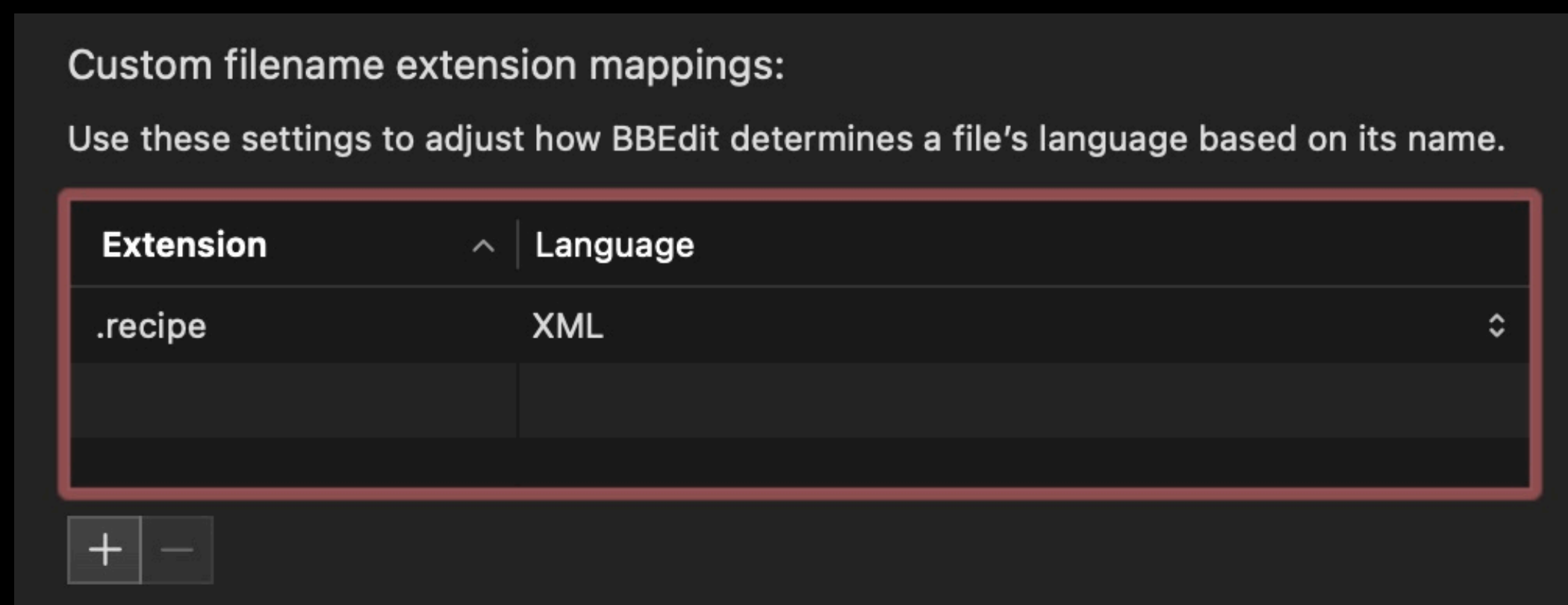
# Recipe Editing Pro Tips

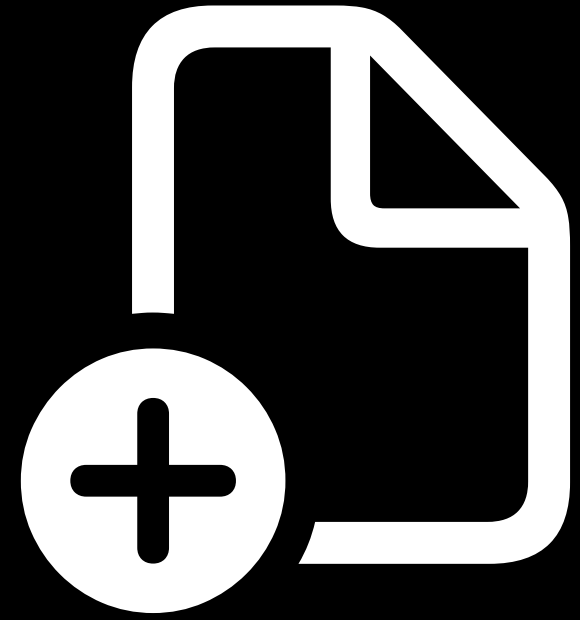
- Choose a supportive text editor
  - ▶ *Syntax highlighting for plist/yaml*

VSCode example settings.json  
file section:

```
"files.associations": {  
  "*.plist": "xml",  
  "*.recipe": "xml",  
  "*.mobileconfig": "xml",  
  "*.entitlements": "xml"  
},
```

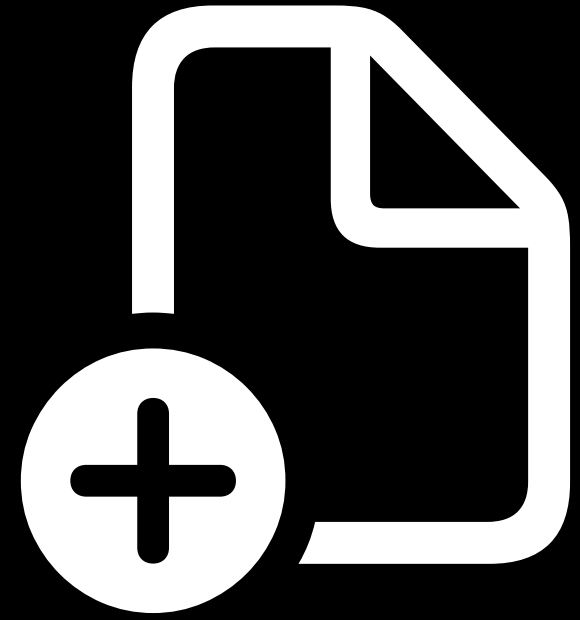
BEdit example Languages setting:





# Recipe Editing Pro Tips

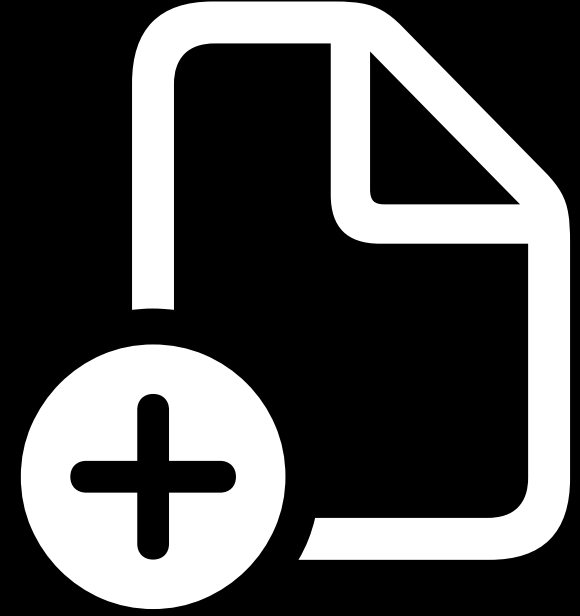
- Choose a supportive text editor
  - ▶ *Syntax highlighting for plist/yaml*
  - ▶ *Autocomplete for plist/yaml*
  - ▶ *Extensible and automatable a plus*
  - ▶ *Some examples: BBEdit, VSCode, Sublime Text, Nova*



# Recipe Editing Pro Tips

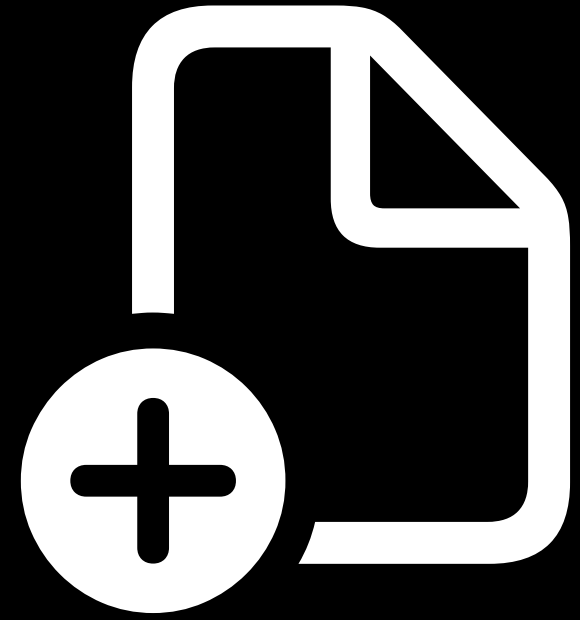
- Reusable snippets to save time
  - ▶ *BEdit: [github.com/jazzace/BEdit-AutoPkg-Clippings](https://github.com/jazzace/BEdit-AutoPkg-Clippings)*
  - ▶ *TextExpander, Typelt4Me, Typinator, Keyboard Maestro, Alfred*





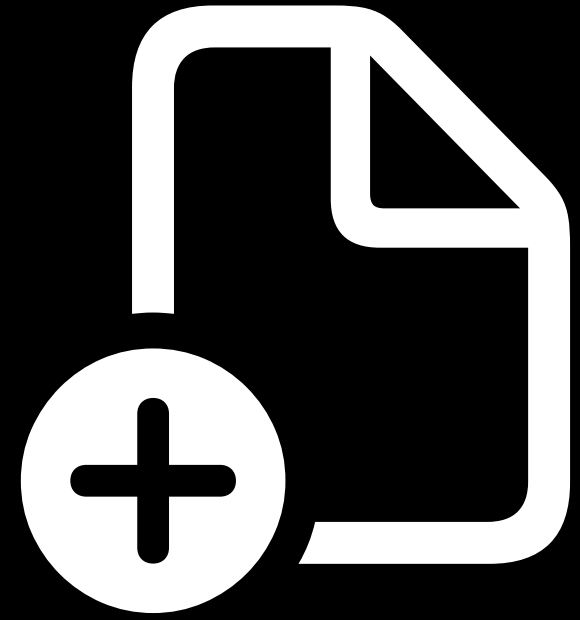
# Recipe Editing Pro Tips

- Use spell check
  - ▶ *cSpell (or an extension that integrates cSpell for your editor)*
  - ▶ *Parses theseWords, these\_words, and these2words*
  - ▶ *Can check an entire folder recursively*



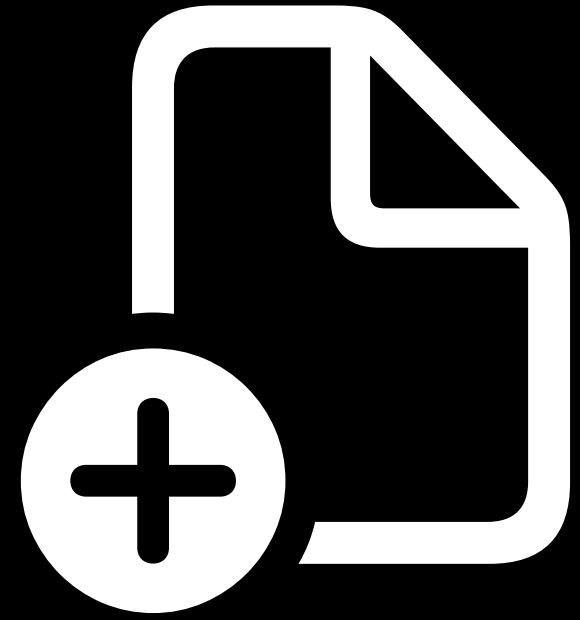
# Recipe Editing Pro Tips

- Autoformat plist files
  - ▶ Always: `plutil -lint *.recipe`
  - ▶ Often: `plutil -convert xml1 *.recipe`
  - ▶ Can be automated upon save or keystroke in many editors
- Autoformat Python files
  - ▶ Python Black
  - ▶ Can also be automated upon save or keystroke



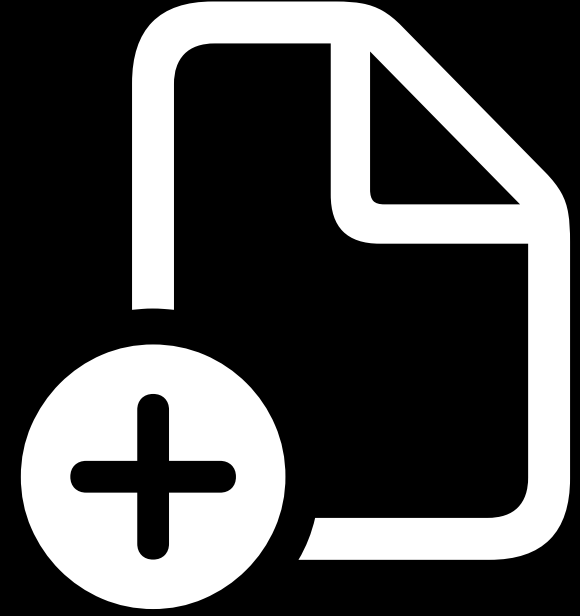
# Recipe Editing Pro Tips

- Pre-commit and CI/CD linters/formatters
  - ▶ <https://github.com/homebysix/pre-commit-macadmin>
  - ▶ *PSU Mac Admins 2019: "Linters, Hooks, and Pipelines - Automation to Save Your Bacon"*



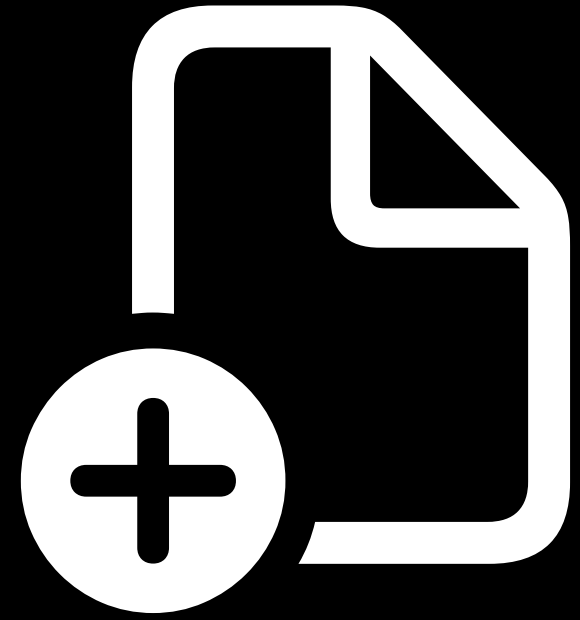
# Recipe Editing Pro Tips

- Assortment of community conventions:
  - ▶ EndOfCheckPhase
  - ▶ Downloading from FossHub (or similar)
  - ▶ Dev feedback about SSL errors
  - ▶ *Vendor feedback about poor scripting*
  - ▶ *Pros/Cons: Versioning in download recipe, or later?*
  - ▶ Don't use %NAME%.app in paths
  - ▶ Recipe Writing Guidelines in Wiki



# URLTextSearcher

- Good to keep as a snippet
- If download URL is dynamic, you can:
  1. *Provide it to Recipe Robot to get the recipes created*
  2. *Edit the download recipe to feed URLDownloader with URLTextSearcher*



# DEMO

- Building a recipe with Recipe Robot





PROBLEMS OUTPUT DEBUG

✔ RecipeRobotOutput 9

about:blank x +

https://raycast.com

Waiting for www.raycast.com...

Elements Console Sources **Network** Performance Memory Application Security Lighthouse Recorder ⌵ Performance insights ⌵

⏏️ 🔍 ☐ Preserve log ☐ Disable cache No throttling ⌵ 📶 ⬆️ ⬇️

Filter ☐ Invert ☐ Hide data URLs **All** Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other ☐ Has blocked cookies ☐ Blocked Requests ☐ 3rd-party requests

20 ms 40 ms 60 ms 80 ms 100 ms 120 ms 140 ms 160 ms 180 ms 200 ms 220 ms 240 ms 260 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
raycast.com	308	document / Redirect	Other	198 B	226 ms	
www.raycast.com	(pending)	document	raycast.com/	0 B	Pending	

2 requests | 198 B transferred | 0 B resources

zsh + 🗑️ ⋮ ⬆️ ⬇️ ✕

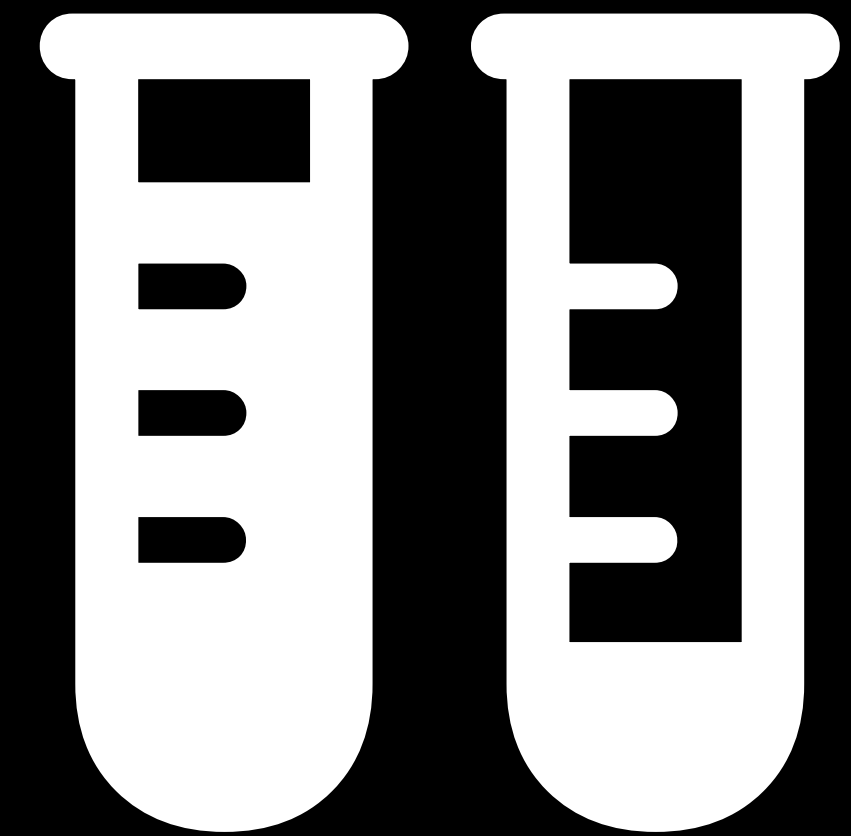


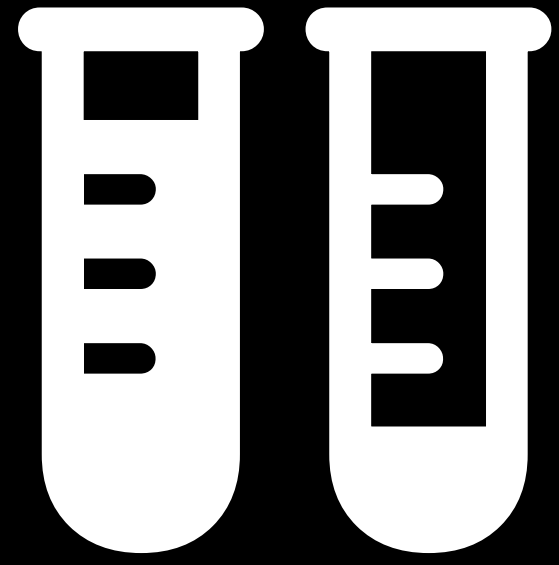
# Your Turn!



- Create an AutoPkg download recipe for any app
  - ▶ *If you don't have an app in mind, try a third-party app installed on your Mac*
  - ▶ *[https://www.producthunt.com/topics/mac?order=most\\_recent](https://www.producthunt.com/topics/mac?order=most_recent)*
- Use any method you like
  - ▶ *Recipe Robot may require stable wi-fi, whereas `new-recipe` verb does not*
  - ▶ *If using Recipe Robot, use `--ignore-existing`*
- When done, a volunteer can walk us through their recipe

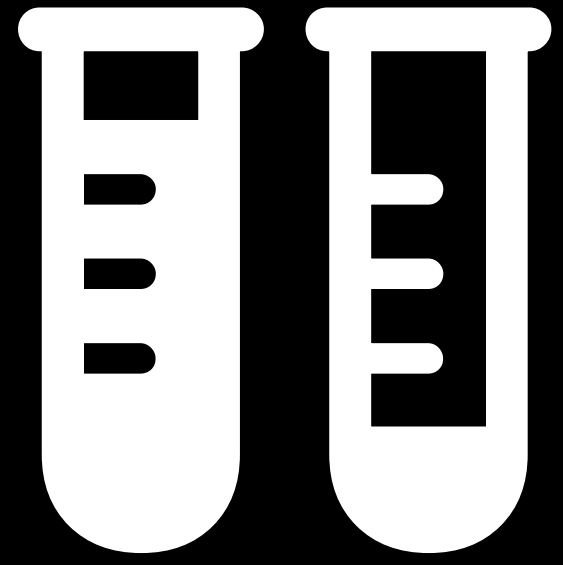
# Testing Your Recipes





# Recipe Search Order

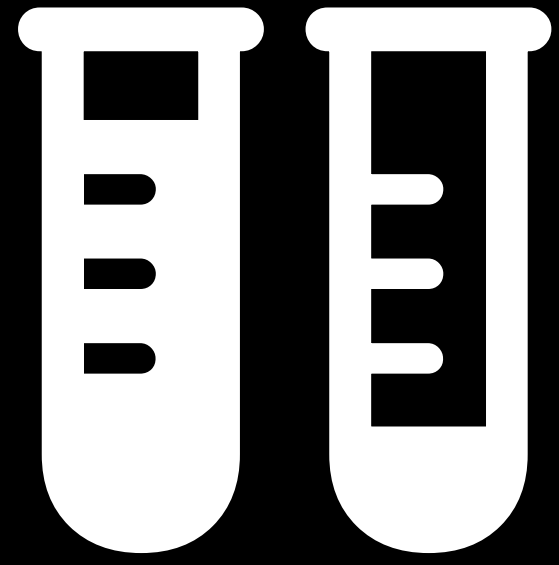
```
% defaults read com.github.autopkg RECIPE_SEARCH_DIRS  
(  
    ".",  
    "~/Library/AutoPkg/Recipes",  
    "/Library/AutoPkg/Recipes",  
    "/Users/areimer/Library/AutoPkg/RecipeRepos/  
com.github.autopkg.recipes",  
    "/Users/areimer/Library/AutoPkg/RecipeRepos/  
com.github.autopkg.jazzace-recipes",  
    "/Users/areimer/Library/AutoPkg/RecipeRepos/  
com.github.autopkg.homebysix-recipes"  
)
```



# Testing 1, 2, 3

(With or without overrides)

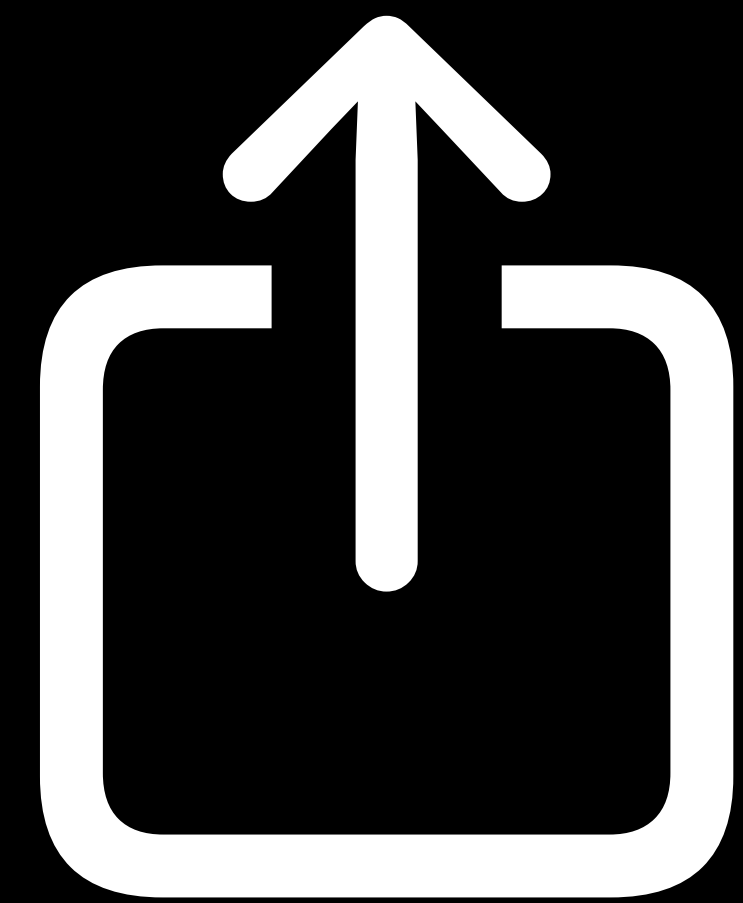
1. Test the parent(s)
2. Test my new child recipe
3. Test it again

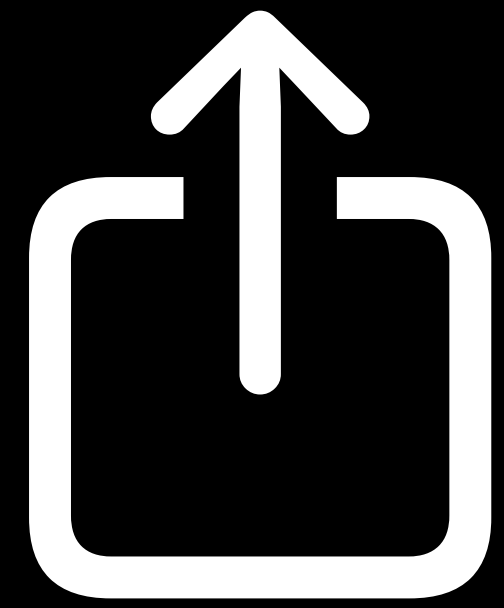


# Testing Tips

- ▶ `autopkg run recipename.type --pkg /path/to/download`
- ▶ `alias apr='autopkg run -vv'`
- ▶ `alias aprt='autopkg run -vv --ignore-parent-trust-verification-errors'`
  - ▶ *Store aliases in .zshrc or .bashrc for persistence*
- Automated tests (neat but limited audience)

# Sharing Your Recipes

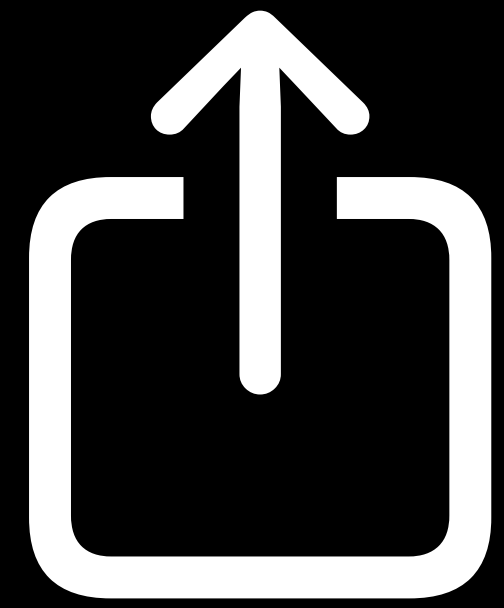




# Sharing Your Recipes

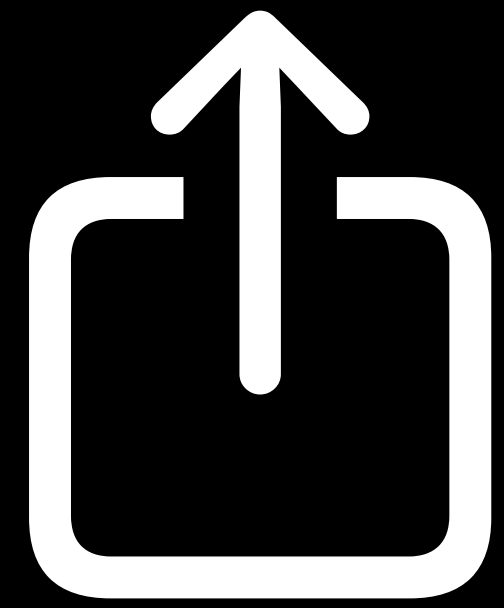
- Why?
  - ▶ *Appear in search results*
  - ▶ *Saves community time*
  - ▶ *Establishes you as recipe author*
- Why not?
  - ▶ *Internal to your company*
  - ▶ *Download behind paywall*
  - ▶ *You don't have time/interest in maintenance*





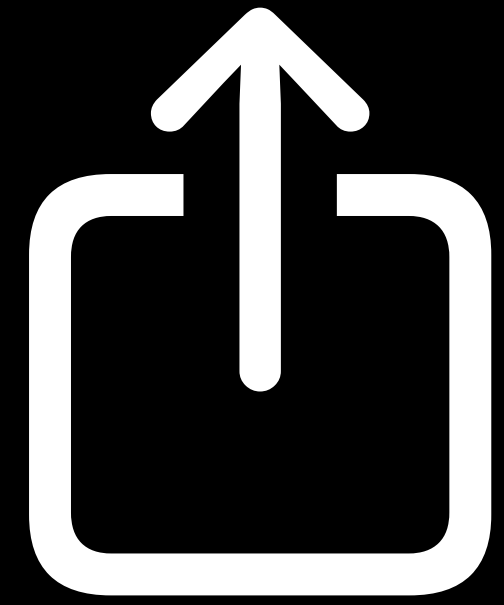
# Sharing Your Recipes

- Decision: PR to existing repo, or whole new repo?
- Create a repo on GitHub and add your recipes
  - ▶ *Subfolders: Vendor vs product name*
  - ▶ *Remember: AutoPkg reads repo root and ONE level down*
- Use repo-add to add your repo and test recipes
- Open an issue to request repo (see wiki) and wait



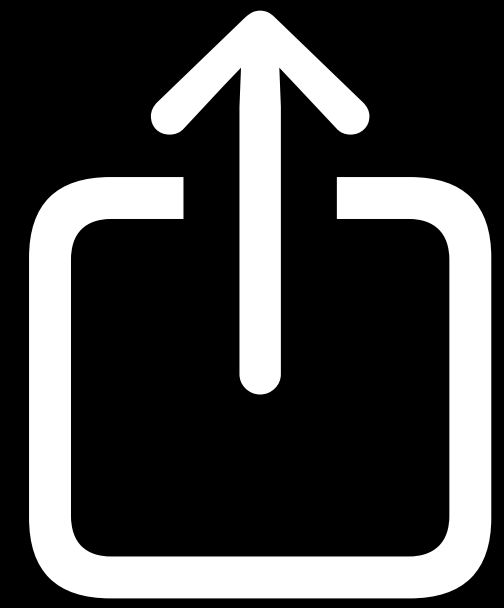
# Sharing Your Recipes

- Recipe maintainer expectations
  - ▶ [github.com/autopkg/autopkg/wiki/Sharing-Recipes#repo-maintenance-expectations](https://github.com/autopkg/autopkg/wiki/Sharing-Recipes#repo-maintenance-expectations)



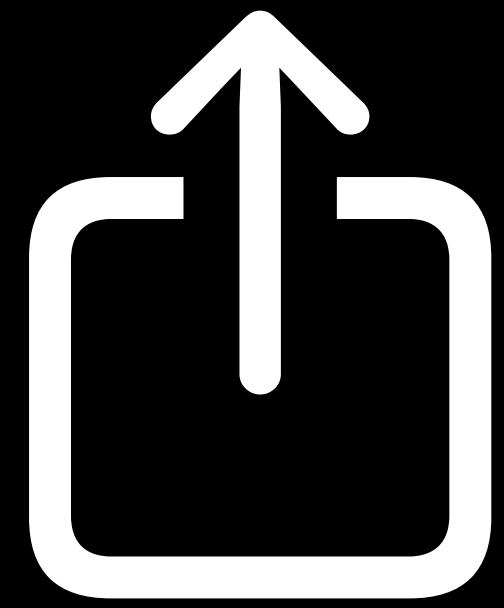
# Sharing Your Recipes

- Principles of excellent recipes:
  - ▶ *Broadly useful*



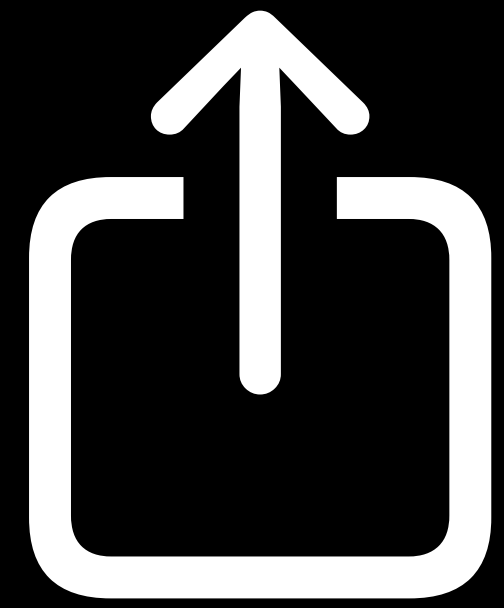
# Sharing Your Recipes

- Principles of excellent recipes:
  - ▶ *Broadly useful*
  - ▶ *Flexible*



# Sharing Your Recipes

- Principles of excellent recipes:
  - ▶ *Broadly useful*
  - ▶ *Flexible*
  - ▶ *Futureproof*



# Sharing Your Recipes

- Principles of excellent recipes:
  - ▶ *Broadly useful*
  - ▶ *Flexible*
  - ▶ *Futureproof*
  - ▶ *Low-maintenance*

# /i( Ongoing Maintenance

- Periodic testing
  - ▶ *Download recipes are most likely to break*
  - ▶ *Test your download recipes periodically*

```
find ~/your/repo -iname "*.download.recipe*" \  
-exec autopkg run -vvcq "{}" '+'
```
  - ▶ *Probably OK to skip Munki, Jamf, and Install recipes*
- Getting help
- Recipe deprecation/ownership transfer
  - ▶ *Tip: DeprecationWarning and change parent to replacement recipe*



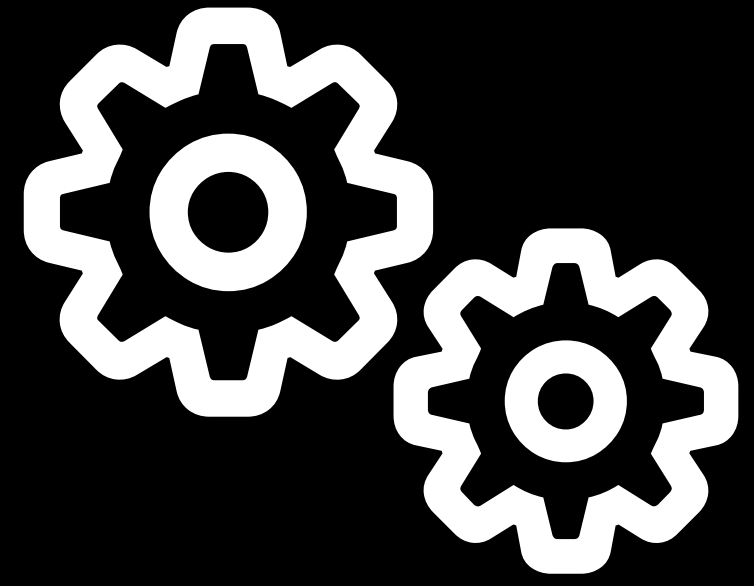
# Outline

- ✓ AutoPkg prerequisites and basics
- ✓ Recipe reading and literacy
- ✓ Running recipes via overrides
- ✓ Specific recipe types
- ✓ Creating new recipes
- ✓ Testing your recipes
- ✓ Sharing your recipes

Advanced topics (you tell us!)

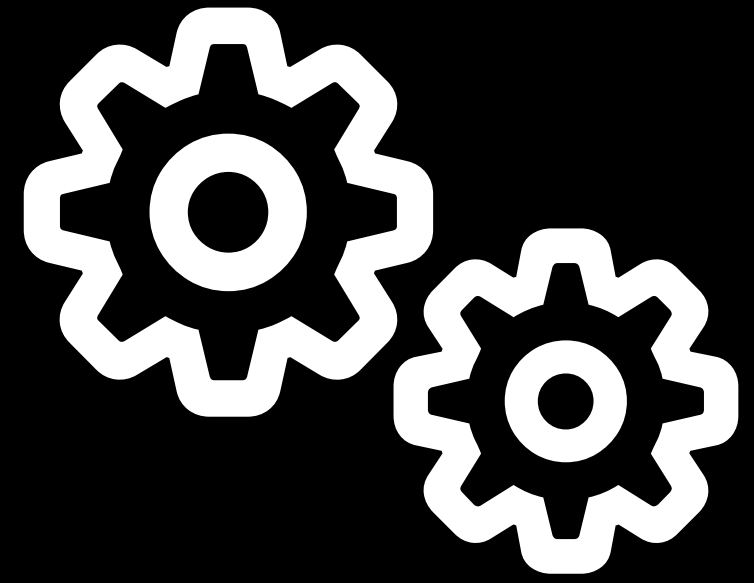
Wrap up

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**



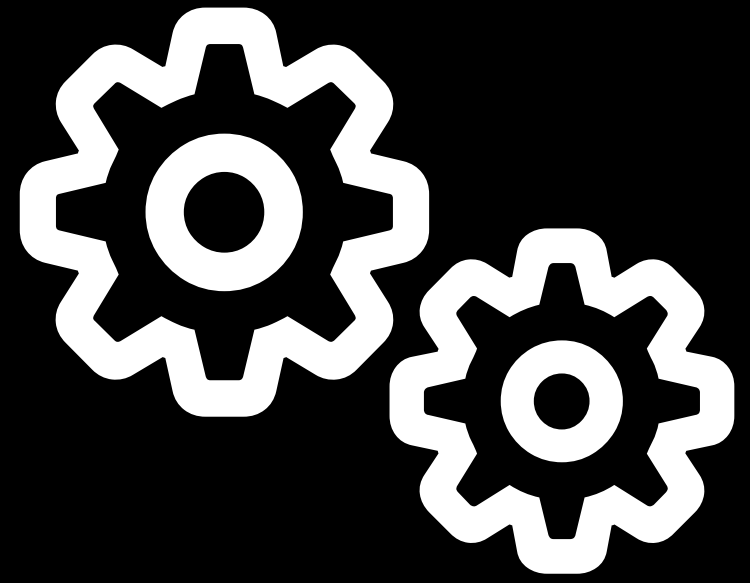
# Processor Anatomy

- Example: MozillaURLProvider
  - ▶ *Shebang*
  - ▶ *License*
  - ▶ *Docstring*
  - ▶ *Imports and \_\_all\_\_*
  - ▶ *Global constants*
  - ▶ *Processor class*
  - ▶ *Input/output variables and description*
  - ▶ *Custom functions*
  - ▶ *Main function*
  - ▶ *If name main...*



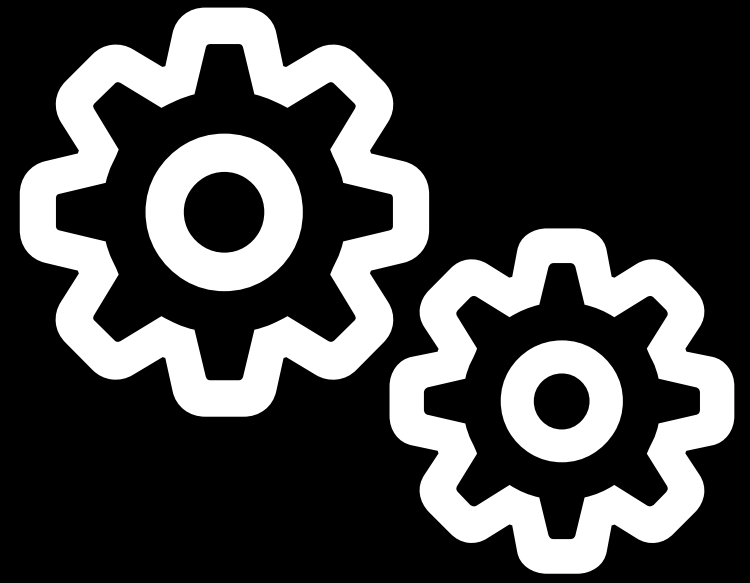
# Processors as Templates

- Tim Sutton's generic starter template
- Duplicate an existing (simple) processor
- Very common: FooURLProvider/FooInfoProvider
- Things to maybe avoid:
  - ▶ *Notarization/signing*
  - ▶ *Credential handling*



# Processor Excellence

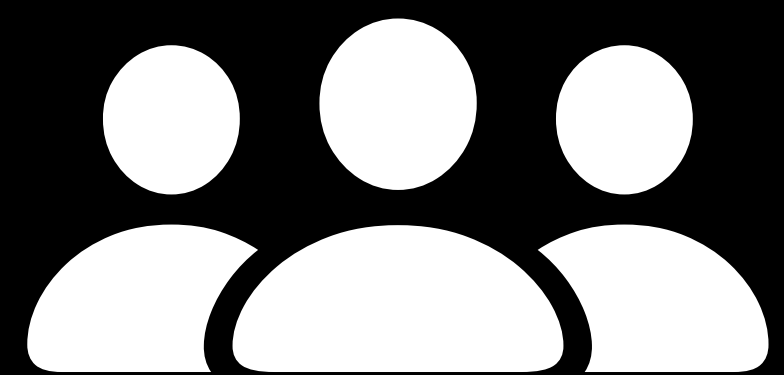
- Broadly useful and flexible
  - ▶ *VendorURLProvider instead of ProductURLProvider*
- Don't assume English
  - ▶ *Add LOCALE input variable if multiple are available*
- Provide useful output variables
- Ensure Python 3 compatibility (RIP Python 2)



# Pro Tip: URLGetter

- Superclass used by processors that download things
- Consider using URLGetter instead of importing urllib or subprocessing curl

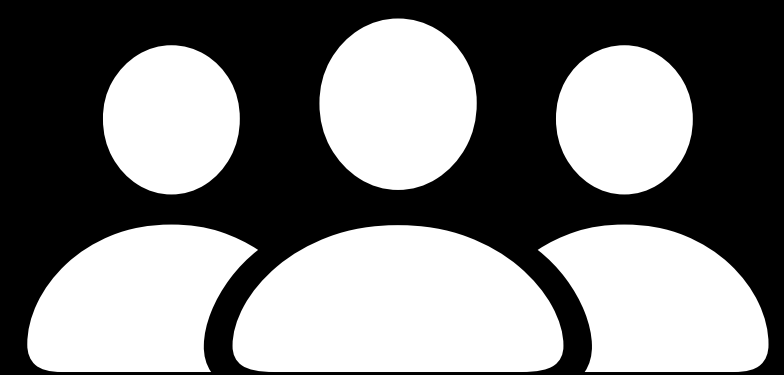
# Contributing to AutoPkg



# AutoPkg Community

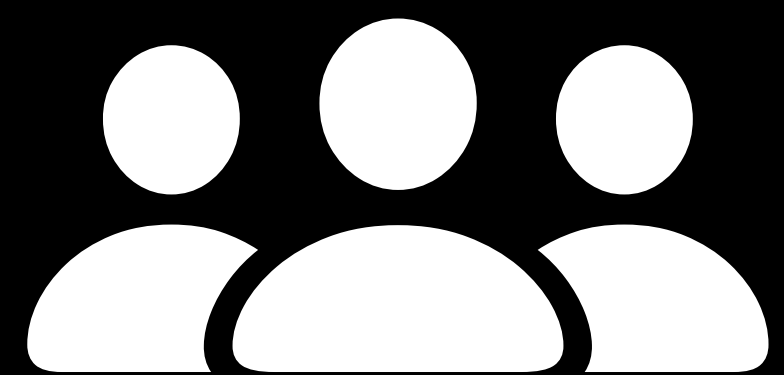
- AutoPkg org on GitHub
  - ▶ *Admins: Greg, Nick, and Elliot are active; a few dormant.*
  - ▶ *Repo Owners: Responsible for merging PRs.*
    - Admins can also merge PRs, but rarely use this power.
  - ▶ *Users: Everyday people opening issues/PRs but who probably don't have a repo of their own.*





# Community Needs

- Reporting/fixing broken recipes
- Contributing/updating documentation
- Testing new AutoPkg versions/features
- Making useful feature requests
- Contributing new recipes



# Issue/PR Etiquette

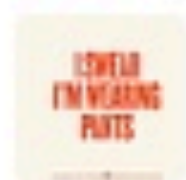
- Many repos have issue templates
- Provide verbose output
  - ▶ *Be sure to conceal any secrets/credentials*
- Provide steps to reproduce



**Sam Schmitt** 11:33 AM

Tuesday, April 21st ▾

I understand why it was done, but having a bunch of commands that start with `list` and then having `repo-list` messes me up every time



**Nick McSpadden**  11:33 AM

same

every time



**Sam Schmitt** 11:34 AM

My fingers instinctively type `list-repos`



**Anthony Reimer** 12:02 PM

I've set up some shell aliases to save me some typing, but my alias building skills are not good enough to properly parse `autopkg list-repos` and have it alias to `autopkg repo-list`, for example. It might be possible, though. (edited)



**elios** 12:05 PM

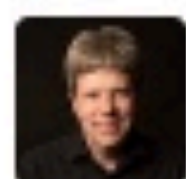
Autopkg could accept both



**Anthony Reimer** 12:05 PM

That would be another solution.

If that was going to be added to the code, it would make sense to offer flipped alternatives for the three verbs that have `list` in them.



**Anthony Reimer** 12:16 PM

I'm looking at the code. Is it as simple as adding the alternate verb(s) to the Main Routine and having them point to the correct function? I don't write in Python but even I could PR that if it is that simple.

<> Code

ⓘ Issues 19

Pull requests 3

▶ Actions

📖 Wiki

🛡 Security 0

📊 Insights

# Add verbs list-repos and processor-list #628

Merged

nmcspadden merged 1 commit into autopkg:dev\_fetch\_parents from jazzace:master 21 days ago

💬 Conversation 2

🔑 Commits 1

📋 Checks 0

📄 Files changed 1

jazzace commented on Apr 21

Member 😊 ⋮

Alternative syntax for repo-list and list-processors

Add verbs list-repos and processor-list ... 0ee0753

Reviewers

No review

Assignees

No one as

Labels



jazzace commented on Apr 21

Member

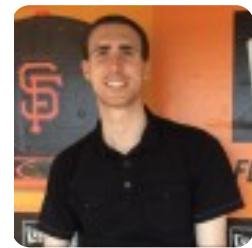


Alternative syntax for repo-list and list-processors



Add verbs list-repos and processor-list ...

0ee0753



nmcspadden commented on Apr 21

Member



At the very minimum, you will need to put more information into your PR. Please show some actual real-world tests that your command works, what the output looks like, etc.



jazzace commented on Apr 21 • edited ▼

Author

Member



No problem. Running in zsh on macOS 10.15.4:

Reviewers

No review

Assignees

No one as

Labels

None yet

Projects

None yet

Milestone

No milesto

Linked iss



# Issue: Bug report

Report an issue with AutoPkg release. If this doesn't look right, [choose a different type](#).

Helpful resources

[Contributing](#)  
[GitHub Community Guidelines](#)



Title

WritePreview

Describe the problem

A clear and concise description of what the problem is.

Preferences contents

*BE SURE TO SANITIZE ANY SENSITIVE DATA SUCH AS PASSWORDS OR ADDRESSES.*  
Provide the output of `defaults read com.github.autopkg` , or the contents of your external `--prefs` file.

AutoPkg output

*BE SURE TO SANITIZE ANY SENSITIVE DATA SUCH AS PASSWORDS OR ADDRESSES.*  
Provide the output of `autopkg run -vvvv <RecipeName>` , or any other command you are running.  
Please include as much data as possible.

Expected behavior

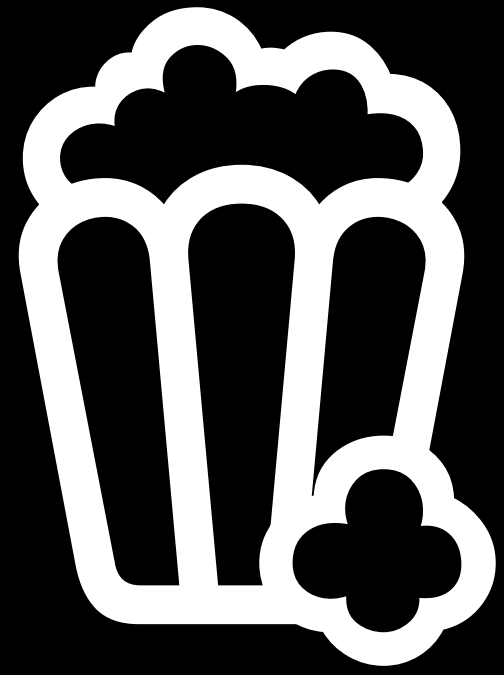
A clear and concise description of what you expected to happen. What specific part of the recipe or AutoPkg run did not behave correctly?

Version (please complete the following information):

- OS version: [e.g. 10.14.6, 10.15.1]
- AutoPkg Version: (generally expected that everyone should be on the latest release, but if you are using master or a specific commit, please specify)

Styling with Markdown is supported

Submit new issue



# Coming Attractions

- Cached-index search
- CI/CD-friendly download metadata
- Windows improvements?
  - ▶ *"Using AutoPkg for Windows Software 2.0"*  
*by James Stewart. Thursday 3:15pm in room 107.*
- *<Your idea here!>*

# Writing Custom Processors







# Custom Advantages

- Can deal with complex/unique download and/or versioning situations
- Customized for that product
- Can be coded to use human-friendly Input values
- Can be more efficient
- Allows addition of features not currently covered by core processors



# Custom Disadvantages

- Requires knowledge of Python to write a custom processor
- User requires knowledge of Python or a good testing scheme to audit a custom processor
- If you can't write Python and the custom processor requires an update, you have to wait for someone else to do it
- More code to maintain



# Is a Custom Processor Needed?

- URLTextSearcher can parse web pages, XML, JSON, even local files
- If combining components into a URL, input variables into URLDownloader arguments
- Multiple core processors vs one custom processor



# Office Recipe Sets

- Core
  - ▶ [\*https://github.com/autopkg/recipes\*](https://github.com/autopkg/recipes)
- Rich Trouton (rtrouton)
  - ▶ [\*https://github.com/autopkg/rtrouton-recipes\*](https://github.com/autopkg/rtrouton-recipes)



# Download & Version

- Core recipes
  - ▶ *URL Source: Microsoft Auto Update (MAU) feed*
  - ▶ *MSOfficeMacURLandUpdateInfoProvider*  
*(URL provider + versioner)*
- Rich Trouton recipes
  - ▶ *URL Source: Fixed link (Microsoft FWLink)*
  - ▶ *Core processors only*



```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Description</key>
    <string>Downloads the latest version of Microsoft's Excel for Office 365 installer package and renames the package.</string>
    <key>Identifier</key>
    <string>com.github.rtrouton.pkg.microsoftexcel365</string>
    <key>Input</key>
    <dict>
      <key>NAME</key>
      <string>Microsoft Excel</string>
      <key>VENDOR</key>
      <string>Microsoft</string>
      <key>SOFTWARETITLE</key>
      <string>Excel</string>
    </dict>
    <key>MinimumVersion</key>
    <string>1.0.0</string>
    <key>ParentRecipe</key>
    <string>com.github.rtrouton.download.microsoftexcel365</string>
    <key>Process</key>
    <array>
      <dict>
        <key>Arguments</key>
        <dict>
          <key>destination_path</key>
          <string>%RECIPE_CACHE_DIR%/downloads/unpack</string>
          <key>flat_pkg_path</key>
          <string>%pathname%</string>
        </dict>
        <key>Processor</key>
        <string>FlatPkgUnpacker</string>
      </dict>
      <dict>
        <key>Arguments</key>
        <dict>
          <key>pattern</key>
          <string>%RECIPE_CACHE_DIR%/downloads/unpack/*Excel*.pkg</string>
        </dict>
        <key>Processor</key>
        <string>FileFinder</string>
      </dict>
      <dict>
        <key>Arguments</key>
        <dict>
          <key>destination_path</key>
          <string>%RECIPE_CACHE_DIR%/downloads/payload</string>
          <key>pkg_payload_path</key>
          <string>%found_filename%/Payload</string>
        </dict>
        <key>Processor</key>
        <string>PkgPayloadUnpacker</string>
      </dict>
      <dict>
        <key>Arguments</key>
        <dict>
          <key>input_plist_path</key>
          <string>%RECIPE_CACHE_DIR%/downloads/payload/Microsoft Excel.app/Contents/Info.plist</string>
          <key>plist_version_key</key>
          <string>CFBundleVersion</string>
        </dict>
        <key>Processor</key>
        <string>Versioner</string>
      </dict>
      <dict>
        <key>Processor</key>
        <string>PkgCopier</string>
        <key>Arguments</key>
        <dict>
          <key>source_pkg</key>
          <string>%pathname%</string>
          <key>pkg_path</key>
          <string>%RECIPE_CACHE_DIR%/VENDOR%_%SOFTWARETITLE%_version%.pkg</string>
        </dict>
      </dict>
      <dict>
        <key>Processor</key>
        <string>PathDeleter</string>
        <key>Arguments</key>
        <dict>
          <key>path_list</key>
          <array>
            <string>%RECIPE_CACHE_DIR%/downloads/unpack</string>
            <string>%RECIPE_CACHE_DIR%/downloads/payload</string>
          </array>
        </dict>
      </dict>
    </array>
  </dict>
</plist>

```

# 1. Run Parent download recipe

► *MicrosoftExcel365.download*

# 2. FlatPkgUnpacker

# 3. FileFinder

► *look for \*Excel\*.pkg inside unpacked installer*

# 4. PkgPayloadUnpacker



# 5. Versioner

# 6. PkgCopier

► *Renames to Microsoft\_Excel\_version*

# 7. PathDeleter

<https://github.com/autopkg/rtrouton-recipes/blob/master/MicrosoftExcel365/MicrosoftExcel365.pkg.recipe>

	<div>core (autopkg/recipes)</div>	<div>rtrouton (autopkg/rtrouton-recipes)</div>
<div>Value for URLDownloader</div>		<div>Input variables</div>
<div>Versioning</div>	<div>Custom Processor</div>	<div> 6 Core Processors</div>

# Outline

- ✓ AutoPkg Prerequisites and Basics
- ✓ Recipe reading and literacy
- ✓ Running recipes via overrides
- ✓ Specific recipe types
- ✓ Creating new recipes
- ✓ Testing your recipes
- ✓ Sharing your recipes
- ✓ Advanced Topics (you tell us!)

Wrap up

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**

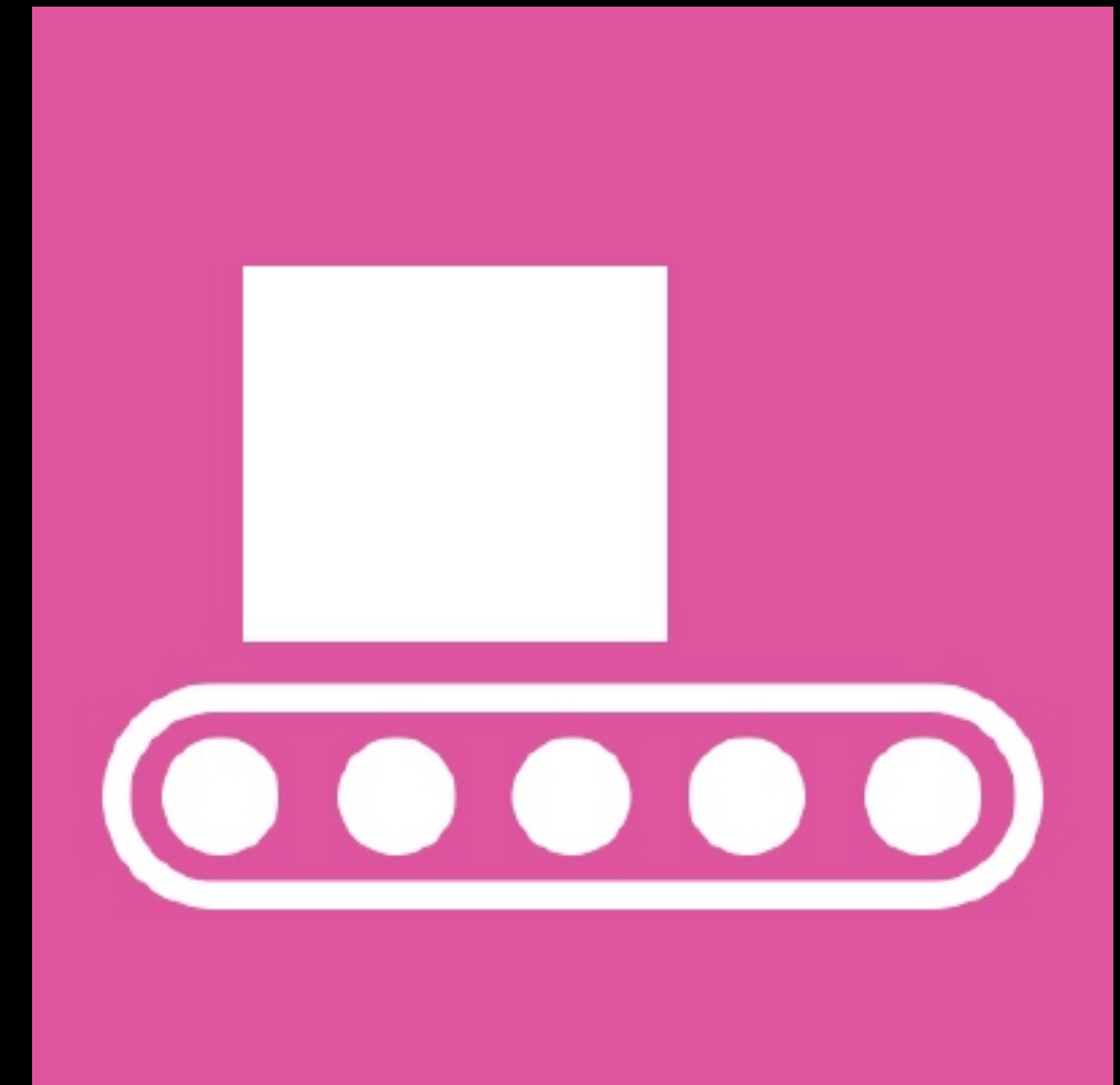


# Wrap Up

# Thank you!

**Feedback:**

**<https://bit.ly/psumac2023-102>**



**Elliot Jordan**

Netflix

 *homebysix*

 *elliotjordan*

 *homebysix*

**Anthony Reimer**

University of Calgary

 *@AnthonyReimer@mastodon.social*

 *jazzace*

 *jazzace*

**Links and References: <https://tinyurl.com/AutoPkgWorkshop>**