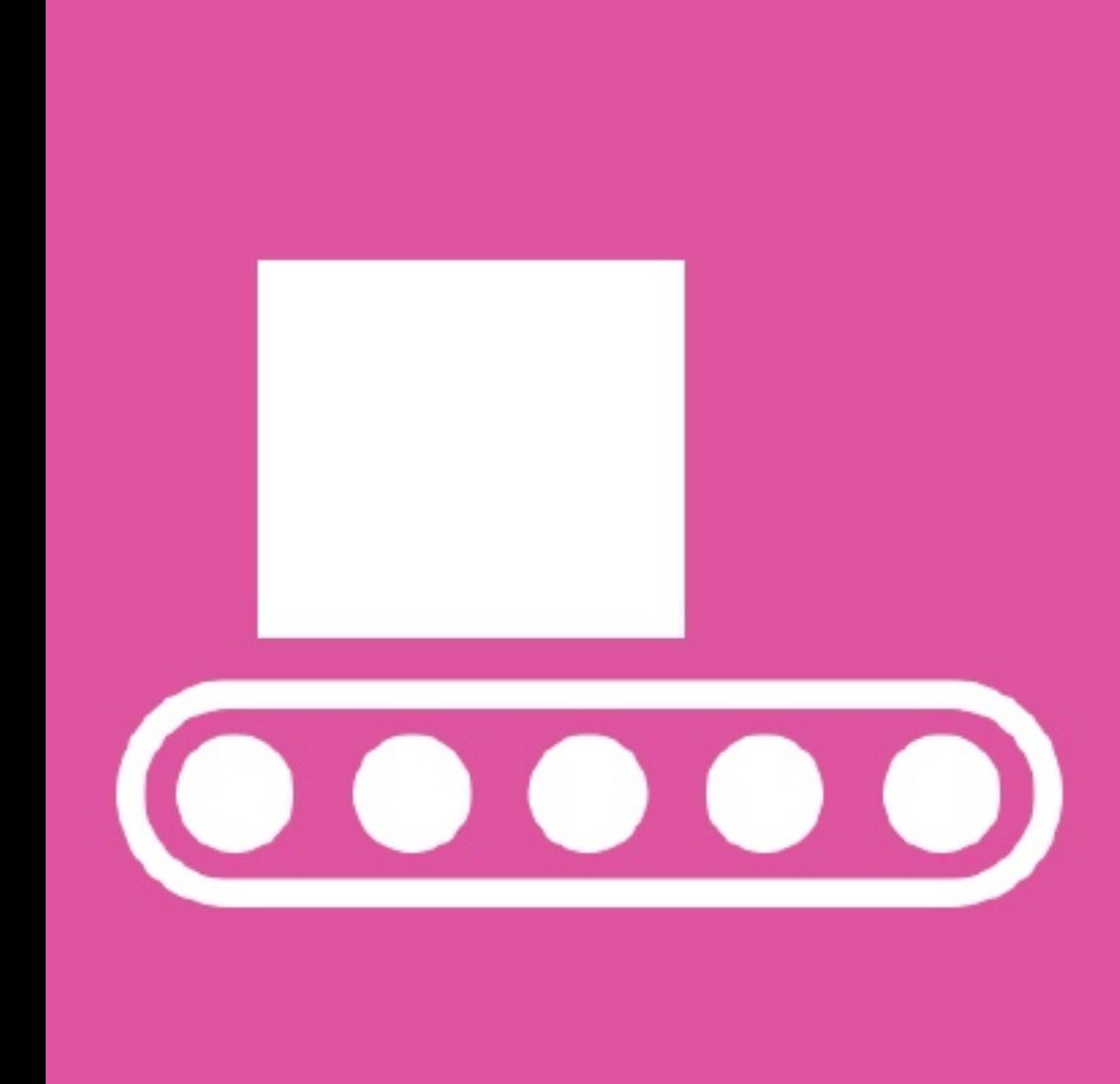


AutoPkg in Depth

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

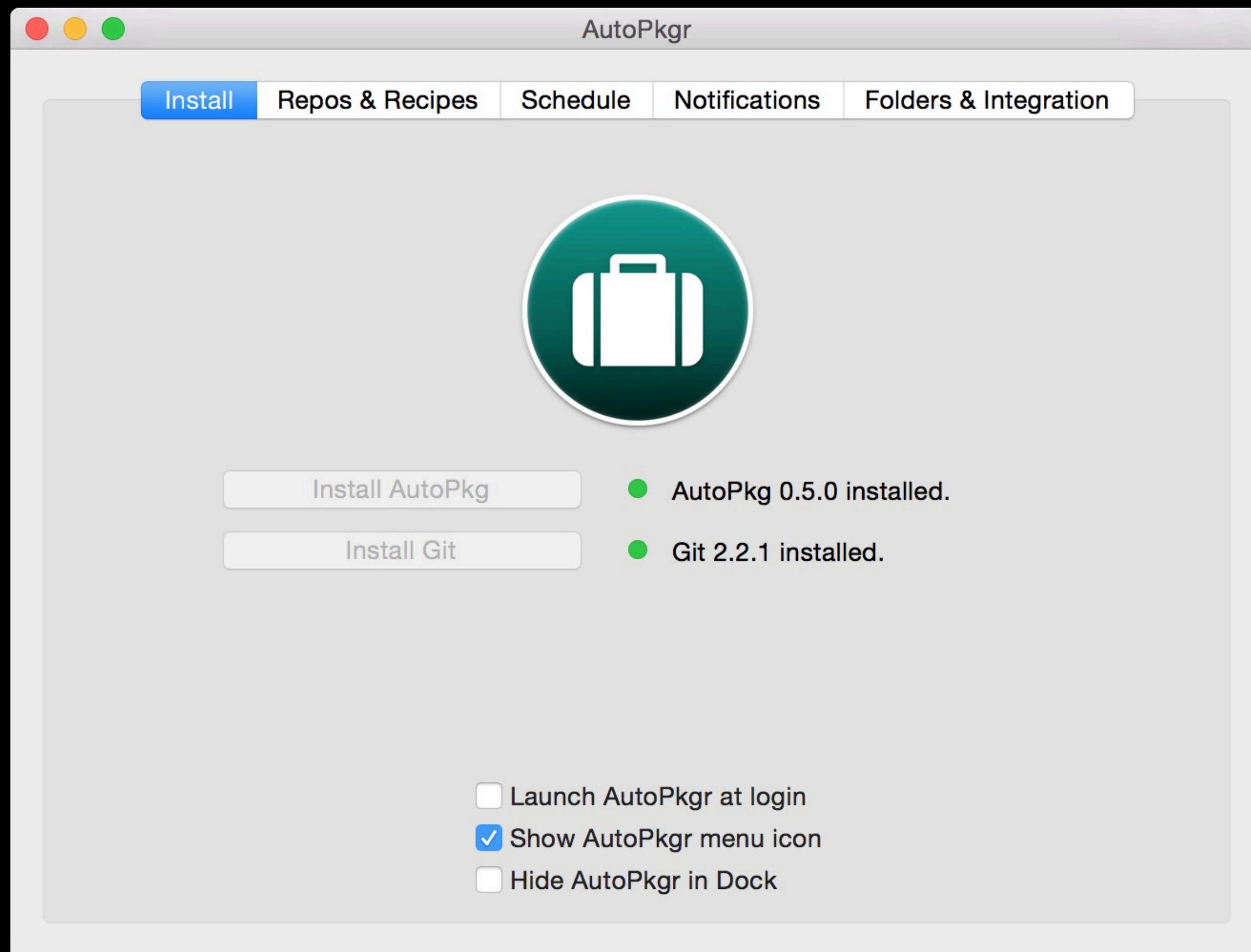
- Requires AutoPkg, Python3, Git
 - ▶ *github.com/autopkg/autopkg/releases*
 - ▶ Xcode Command Line Tools:
developer.apple.com/download/
- Automated installation:
 - ▶ *AutoPkgSetup – Graham Pugh*
github.com/grahampugh/AutoPkgSetup
 - ▶ *AutoPkg*

Standard Workflow

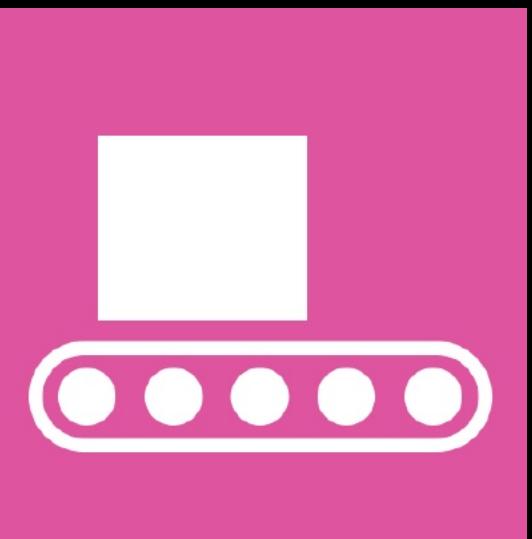
1. Search for recipe(s)
2. Add recipe repo(s)
 - ▶ *Include repos of parent recipes as needed*
3. Create override(s)
4. Run the override(s)
5. Maintain
 - ▶ *Update repos and override trust info*



AutoPkgr



- GUI front end for AutoPkg
 - ▶ github.com/lindegroup/autopkgr



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
$\frac{1}{4}$ tsp.	salt	1 mL
$\frac{1}{2}$ cup	sugar	125 mL
3 tbsp.	cocoa	50 mL
$\frac{1}{2}$ cup	chopped nuts	125 mL
$\frac{1}{2}$ cup	milk	125 mL
1 tsp.	vanilla	125 mL
2 tbsp.	melted butter or margarine	5 mL
		25 mL

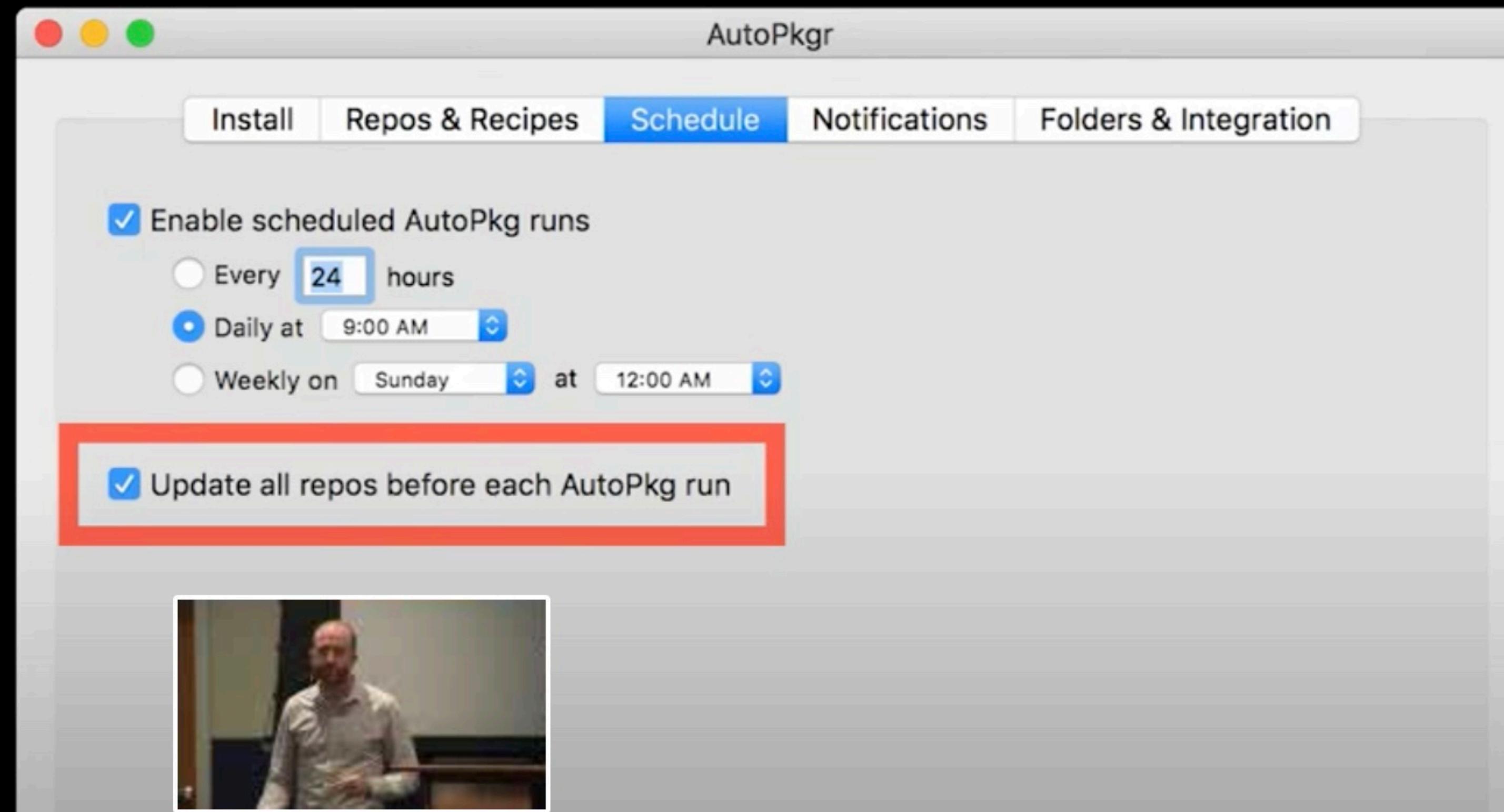
Sauce

4 tbsp.	cocoa	65 mL
$\frac{1}{2}$ cup	sugar	125 mL
2 cups	hot water	500 mL

Sift into a two quart (litre) casserole, flour, baking powder, salt, $\frac{1}{2}$ 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 $\frac{1}{2}$ 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

Identifier
Description

Input

Process

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

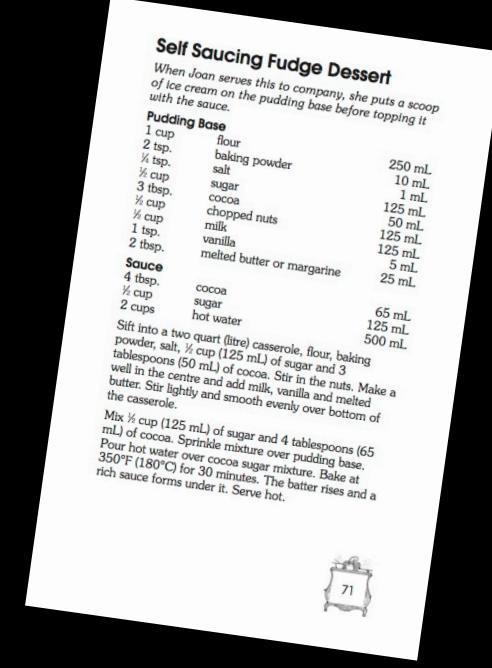
1 cup	flour	250 mL
2 tsp.	baking powder	10 mL
$\frac{1}{4}$ tsp.	salt	1 mL
$\frac{1}{2}$ cup	sugar	125 mL
3 tbsp.	cocoa	50 mL
$\frac{1}{2}$ cup	chopped nuts	125 mL
$\frac{1}{2}$ cup	milk	125 mL
1 tsp.	vanilla	5 mL
2 tbsp.	melted butter or margarine	25 mL

Sauce

4 tbsp.	cocoa	65 mL
$\frac{1}{2}$ cup	sugar	125 mL
2 cups	hot water	500 mL

Sift into a two quart (litre) casserole, flour, baking powder, salt, $\frac{1}{2}$ 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 $\frac{1}{2}$ 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.



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



Supported by AutoPkg



Supported by AutoPkgr

9000

Moderate

Recipes Available

~500

Easy

Readability



Can chain with other format



YAML

✓ (2.3+)



~500

Code

Issues 8

Pull requests 3

Actions

Projects

Wiki

Security

Insights

main

2 branches

0 tags

Go to file

Add file

Code

About

AutoPkg recipes

[Readme](#)[Apache-2.0 license](#)[63 stars](#)[14 watching](#)[35 forks](#)[Report repository](#)

Releases

No releases published

Packages

No packages published

Contributors 33

[+ 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>
    Description
    <key>Identifier</key>
    <string>com.github.yourgithubusername.pkg.nothing</string>
    Identifier
    <key>Input</key>
    <dict>
        <key>NAME</key>
        <string>nothing</string>
    </dict>
    Input
    <key>MiniumumVersion</key>
    <string>1.0</string>
    MinimumVersion
    <key>ParentRecipe</key>
    <string>com.github.yourgithubusername.download.nothing</string>
    ParentRecipe
    <key>Process</key>
    <array>
        <dict>
            <key>Arguments</key>
            <dict>
                <key>Argument1</key>
                <string>Value1</string>
            </dict>
            <key>Processor</key>
            <string>AppPkgCreator</string>
        </dict>
        Processor(s)
    </array>
    Process
</dict>
</plist>
```

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'. At the top, there's a pink square icon with a white square and four dots below it, followed by the organization name 'autopkg'. Below the header, there are tabs for Overview, Repositories (156), Projects, Packages, Teams (4), and People (167). A 'Follow' button is in the top right corner.

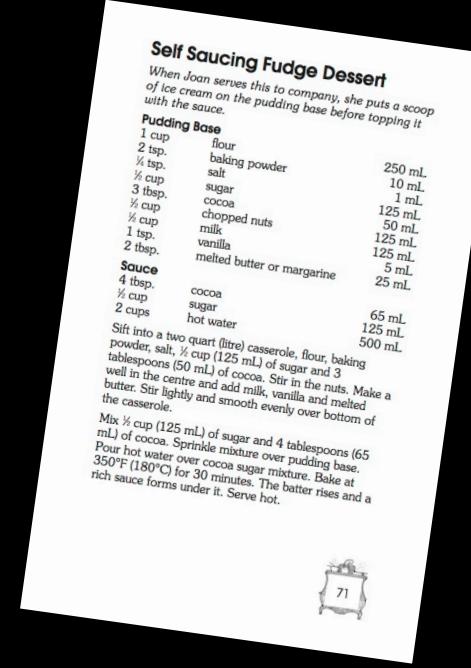
The main area is titled 'Pinned' and contains four repository cards:

- autopkg** (Public) - Automating packaging and software distribution on macOS. Python, 1.2k stars, 192 forks, 48 issues, 7 pull requests, updated 2 days ago.
- homebysix-recipes** (Public) - AutoPkg recipes all the way from Seattle, WA. Shell, 133 stars, 111 forks, 6 issues, 2 pull requests, updated last week.
- n8felton-recipes** (Public) - Recipes for Autopkg. Python, 27 stars, 34 forks, 28 issues, 6 pull requests, updated 2 weeks ago.
- recipes** (Public) - Recipes for AutoPkg. Python, 214 stars, 200 forks, 28 issues, 6 pull requests, updated 2 weeks ago.

Below the pinned section is a search bar with 'Find a repository...' placeholder, and filters for Type, Language, and Sort.

The main repository list starts with 'autopkg' again, followed by 'recipes', 'homebysix-recipes', 'rtrouton-recipes', and 'dataJAR-recipes'. Each card includes a green line graph representing activity over time.

On the right side, there are sections for 'People' (a grid of user icons), 'Top languages' (Python, Shell), and 'Most used topics' (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
.jamf
.filewave
.install

Adjectives

Parent

Child

Override

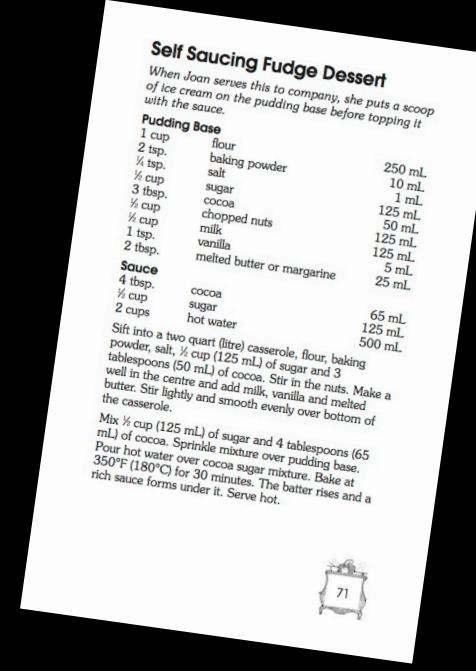
Stub

plist

YAML

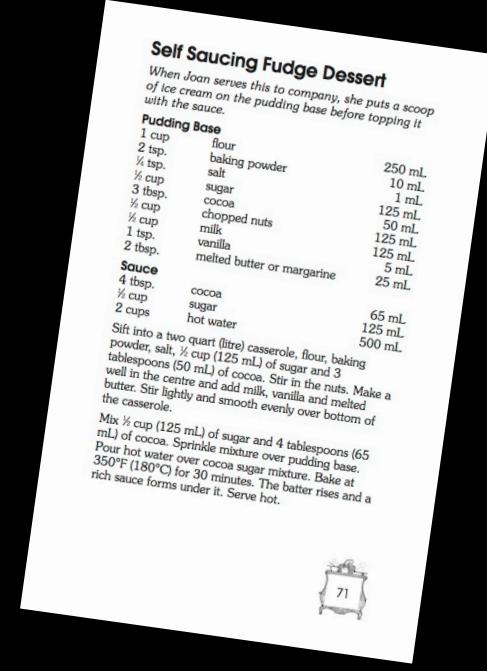


They're all just *recipes*



By Relationship (Kind)

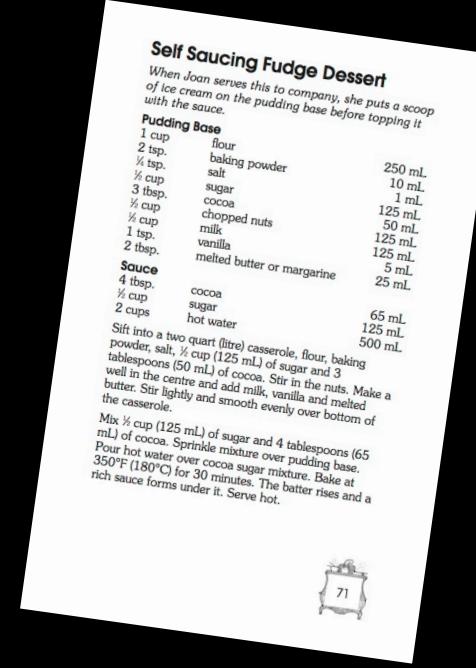
- Parent/Child
- Override
 - ▶ *Created/stored locally*
- Stub
 - ▶ e.g., *homebysix-recipes/FindAndReplace/FindAndReplace.recipe*
- Template
 - ▶ e.g., *jazzace-recipes/JamfTemplates/RecipeTemplates*



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.

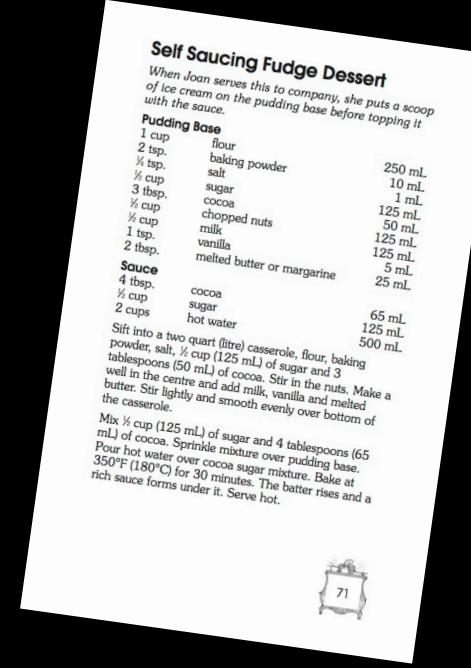


Variables

github.com/autopkg/autopkg/wiki/Variables

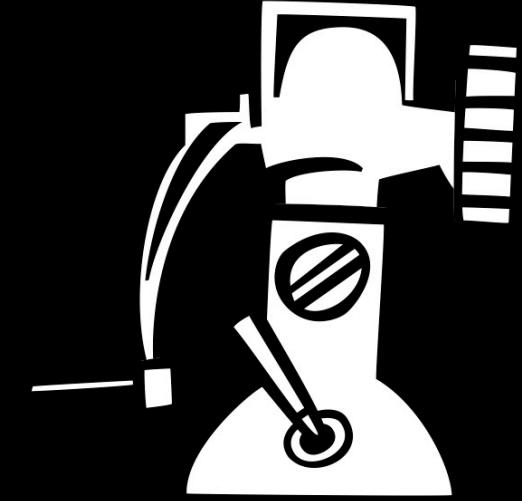
Set via:

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



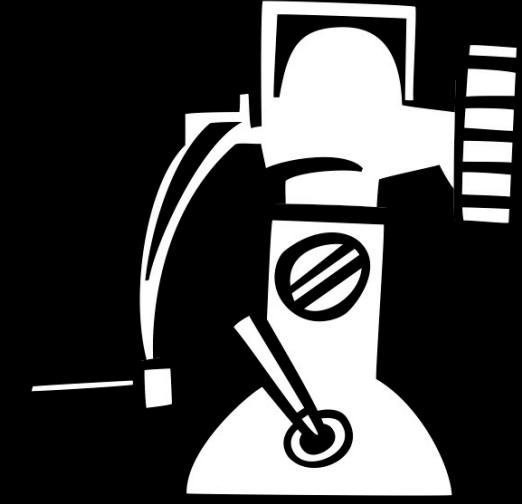
Variables

- 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)
- Variables defined at runtime
 - ▶ *Variable substitution (%VARNAME%)*
 - ▶ *And why some processors fail (uncommon)*



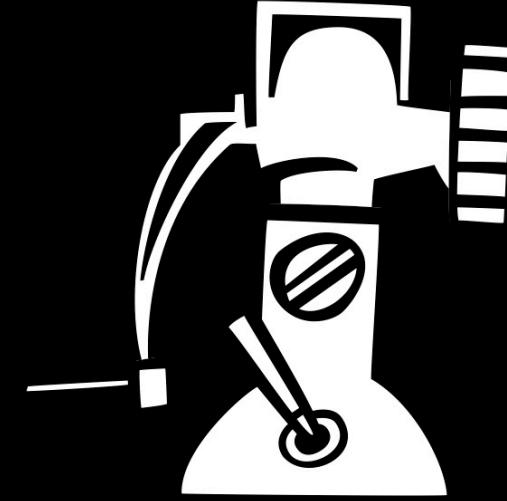
Processors (conceptually)

- 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

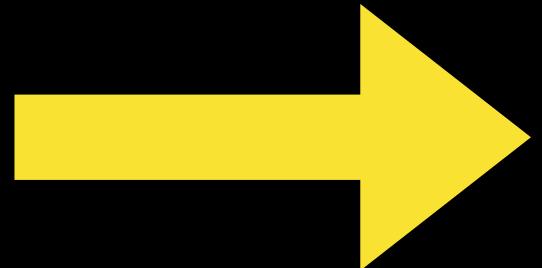


SparkleUpdateInfoProvider

Required Input Variable (Argument)

- appcast_url

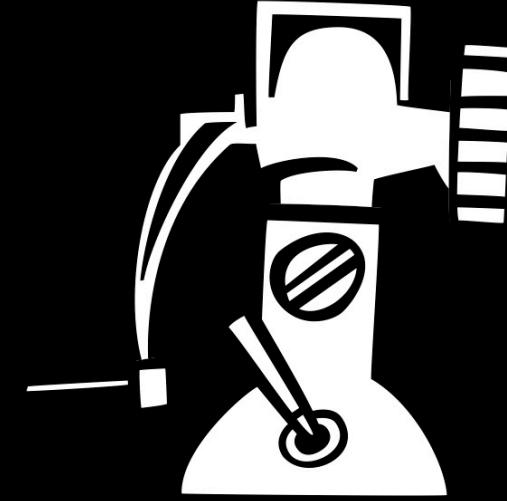
Output Variables



- url
- version
- additional_pkginfo

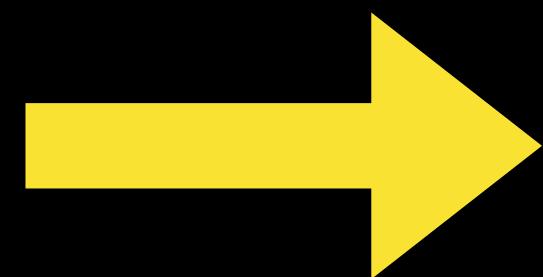
```
[ ... ]  
<key>Process</key>  
<array>  
  <dict>  
    <key>Processor</key>  
    <string>SparkleUpdateInfoProvider</string>  
    <key>Arguments</key>  
    <dict>  
      <key>appcast_url</key>  
      <string>https://ggnome.com/appcast/pano2vr/stable</string>  
    </dict>  
  </dict>  
  <dict>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

<https://github.com/autopkg/jazzace-recipes/blob/master/GardenGnome/Pano2VR.download.recipe>



URLDownloader

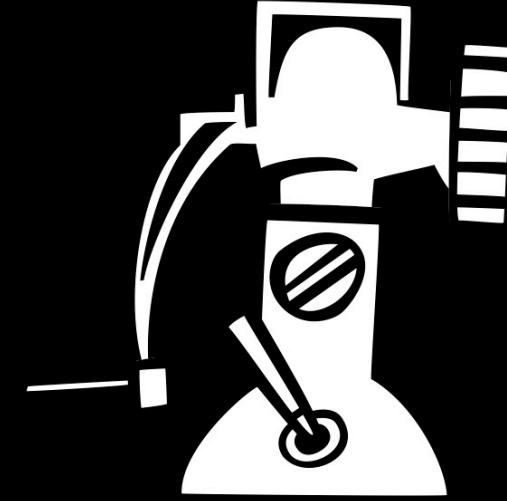
Required Input Variable (Argument)



- url

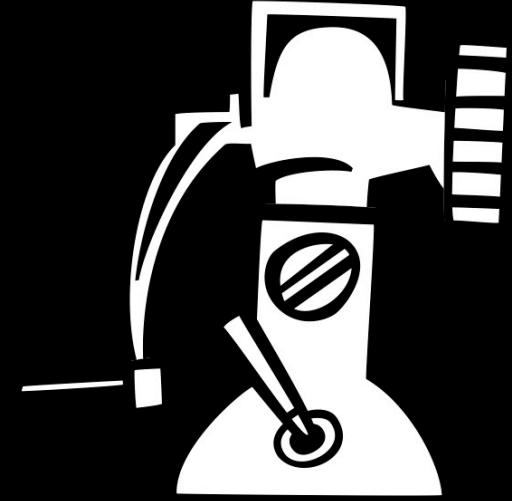
Output Variables

- pathname
- last_modified
- url_downloader_summary_result
- etag
- download_changed



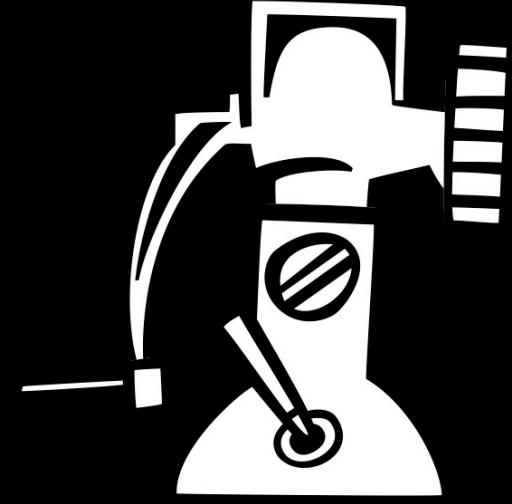
Core Processors can...

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



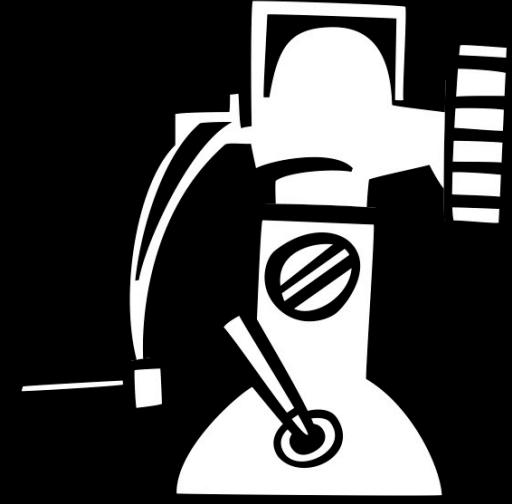
Core Processors can...

- 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*, *MunkiInstallsItemsCreator*, *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

<https://github.com/autopkg/autopkg/wiki/Processor-URLDownloader>

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](#)

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.

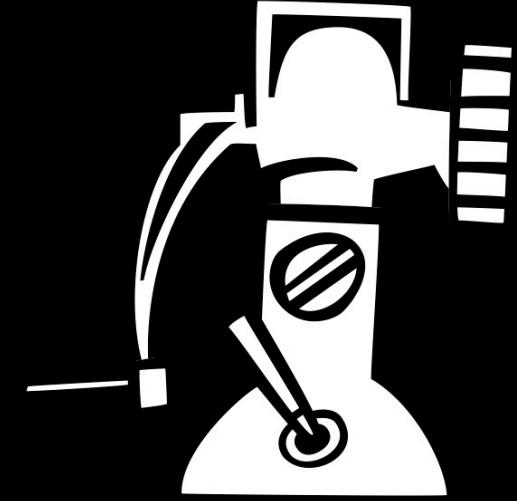
33 + (Can also be used to create directory structures for other purposes.)"""

33 34 description = __doc__

34 35 input_variables = {

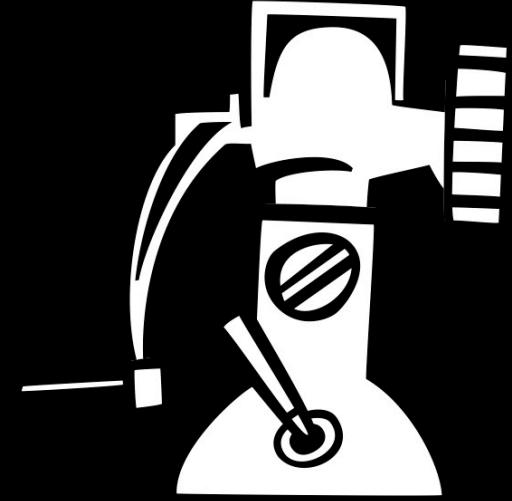
35 36 "pkgroot": {

....



Custom Processors

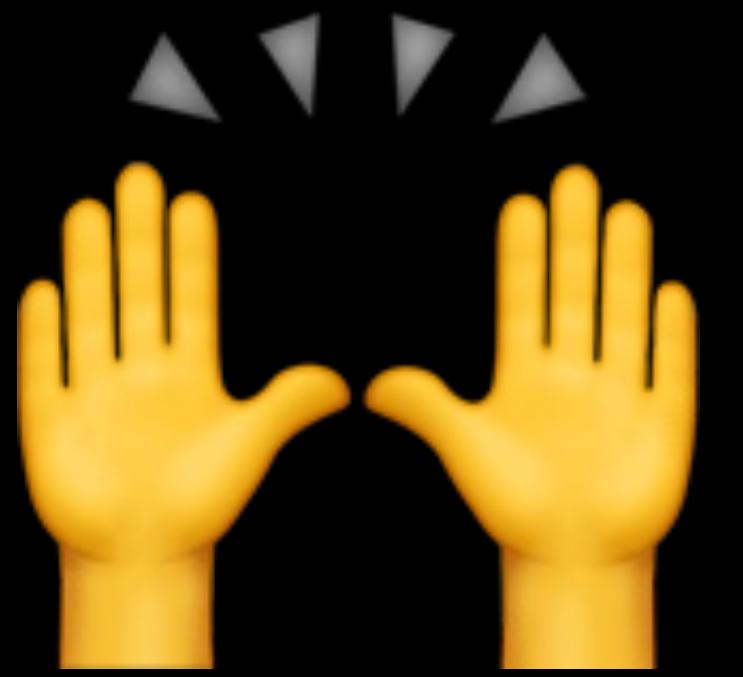
- Perform tasks not covered by Core processors
- Maintained by an individual (or organization)
- Less trusted
- Kept with recipe that uses it

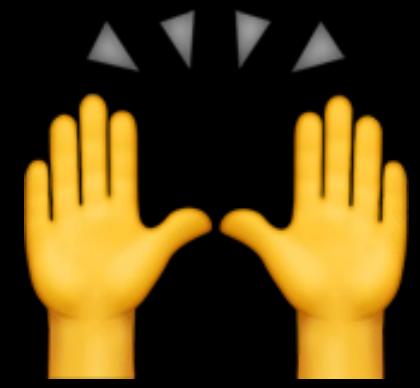


Shared Processors

- 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



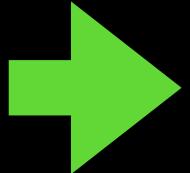
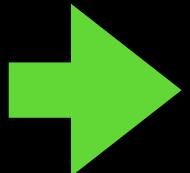


Searching for Recipes

- `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`, then Search within org
 - ▶ <https://github.com/search?type=code&q=org%3Aautopkg+firefox>

```
% autopkg search Firefox | grep pkg\.\.recipe  
MultiFirefox.pkg.recipe ahousseini-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 ahousseini-recipes
EMCSynclicity.pkg.recipe bkerns-recipes
EMCSynclicitySSO.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



Search: org:autopkg Firefox AND .pkg.recipe

Filter by

- Code 21
- Repositories 0
- Issues 1
- Pull requests 1
- Discussions 0
- Users 0
- More

Languages

- XML
- YAML
- Markdown
- More languages...

Repositories

21 files (168 ms) in autopkg X

autopkg/recipes · Mozilla/Firefox.pkg.recipe

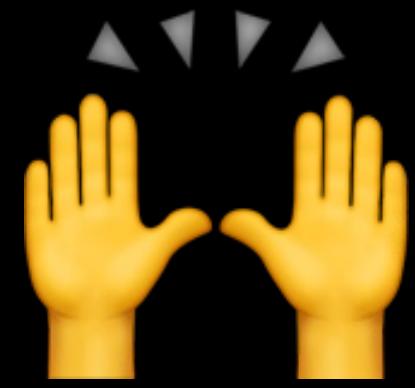
```
5     <key>Description</key>
6     <string>Downloads Firefox disk image and builds a package.
7     Some useful values for RELEASE are: 'latest', 'esr-latest', 'beta-latest'.
10    See the following URLs for more info:
11    http://ftp.mozilla.org/pub/firefox/releases/latest/README.txt
12    http://ftp.mozilla.org/pub/firefox/releases/latest-esr/README.txt
```

Show 7 more matches

autopkg/smithjw-recipes · Mozilla/Mozilla_Firefox.pkg.recipe.yaml

```
1 Description: Downloads the latest version of Firefox and renames it with t
2 Identifier: com.github.smithjw.pkg.Mozilla_Firefox
3 ParentRecipe: com.github.autopkg.download.FirefoxSignedPkg
4 MinimumVersion: '2.3'
5
6 Input:
7 NAME: 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 `plistyaml plist` 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

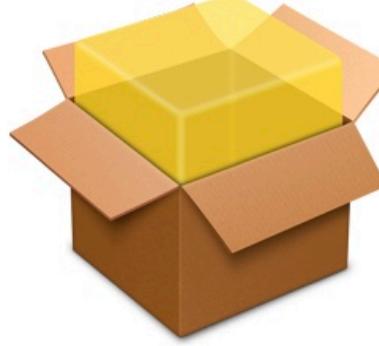
PkgCreator

Firefox-114.0.2.p... Back/Forward Path Action Get Info Quick Look Installer Search 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 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

Mozilla-Firefox-1... Back/Forward Path Action Get Info Quick Look Installer Search Export 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 [?](#)

Mozilla-Firefox-1... Back/Forward Path Action Get Info Quick Look Installer Search Export 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



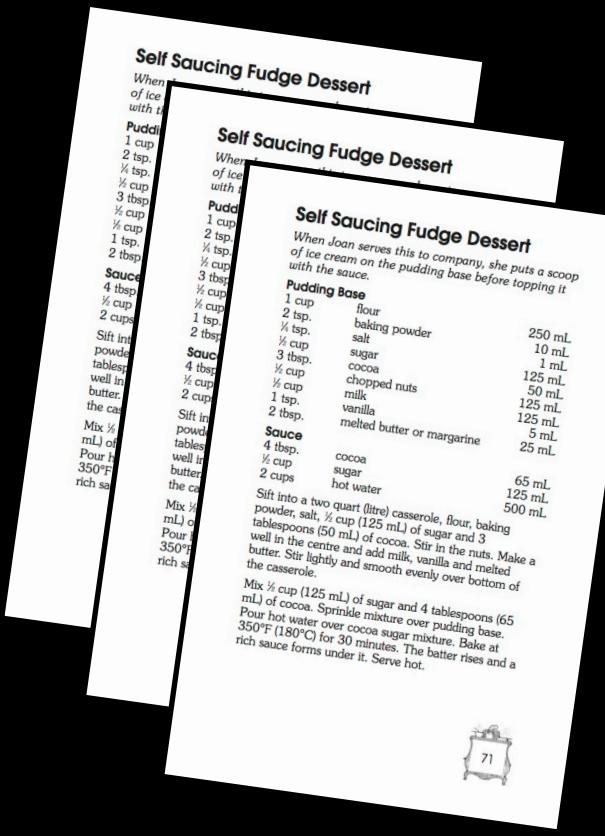
recipes

vs.



rtrouton-recipes

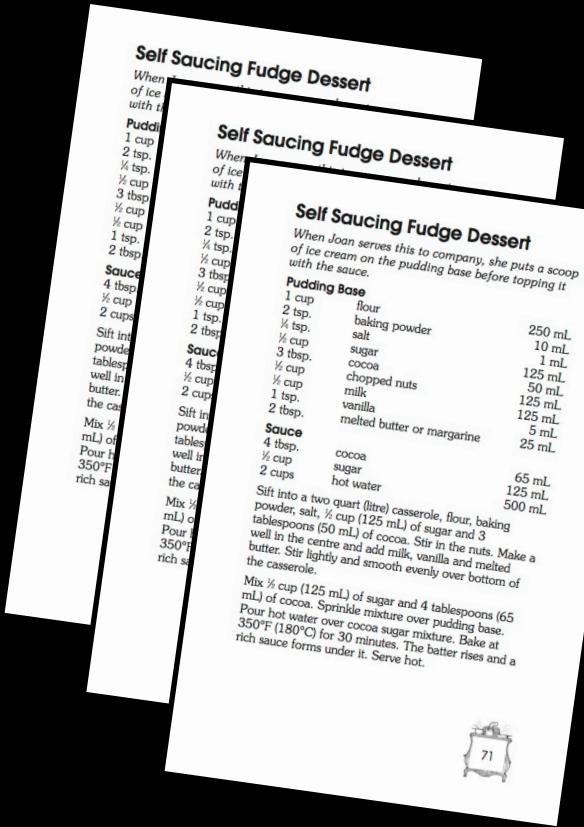
Why Duplicates?



In order of most to least common reason:

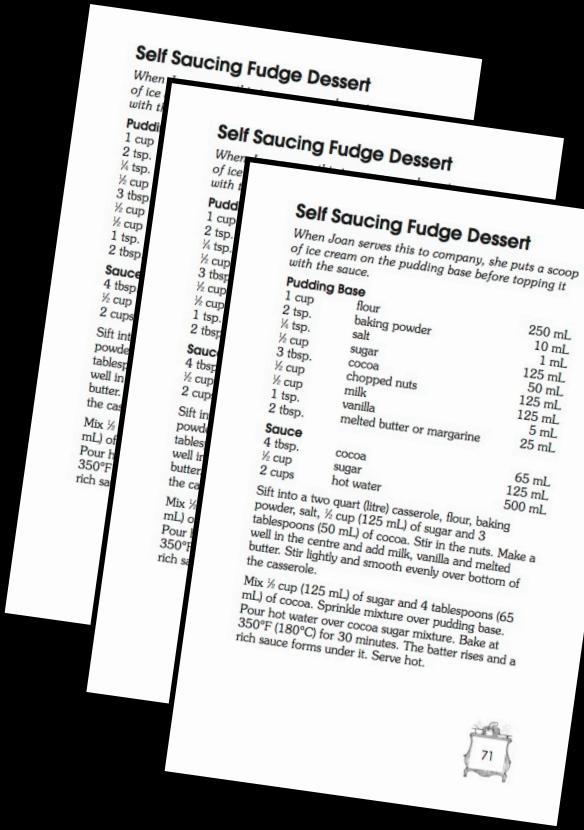
- Specific need
- Existing recipe broken
- Major Version vs. Universal
- Unaware of cross-repo parent-child

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

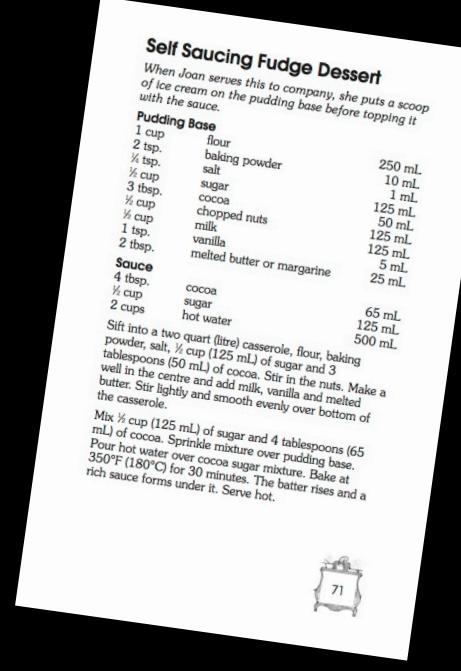
Links and References: <https://tinyurl.com/AutoPkgWorkshop>

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

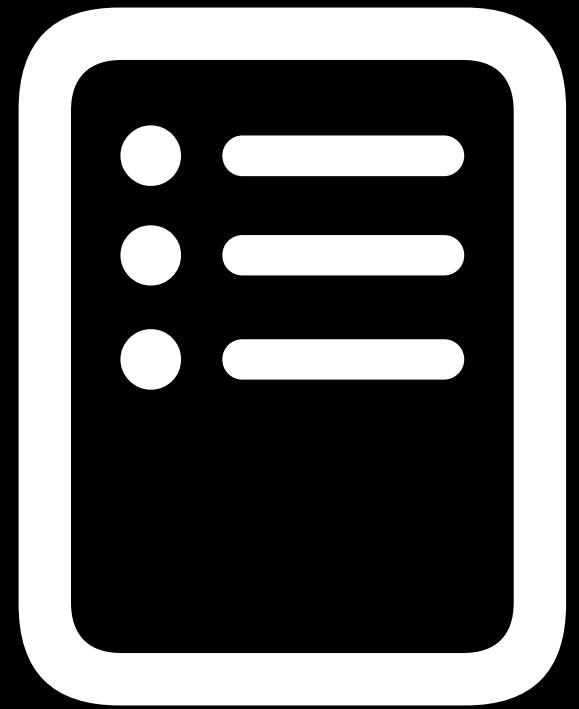
- ▶ *Created/stored locally*

- Stub

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

- Template

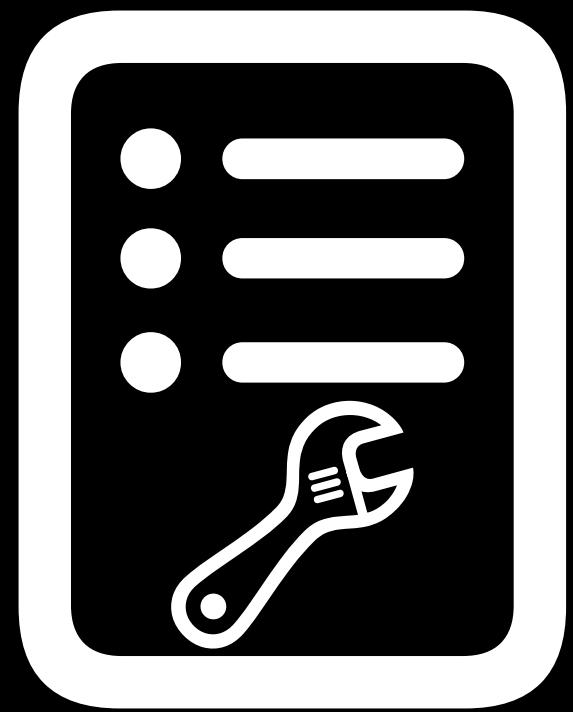
- ▶ e.g., *jazzace-recipes/JamfTemplates/RecipeTemplates*



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>
    Identifier (typically starts with local.)
    <key>Input</key>
    <dict>
        <key>NAME</key>
        <string>nothing</string>
    </dict>
    Input
    <key>ParentRecipe</key>
    <string>com.github.yourgithubusername.pkg.nothing</string>
    ParentRecipe
    <key>ParentRecipeTrustInfo</key>
    <dict>
        <key>non_core_processors</key>
        <dict/>
        non_core_processors
        <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>
            ParentRecipeTrustInfo
            <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>
            parent_recipes
        </dict>
    </dict>
    <dict>
        <key>Identifier</key>
        <string>local.pkg.nothing</string>
    </dict>
</dict>
</plist>
```



Variables in Overrides

- Example: Intel vs Apple Silicon Architecture

GitHubDesktop.download.recipe
(parent of parent)

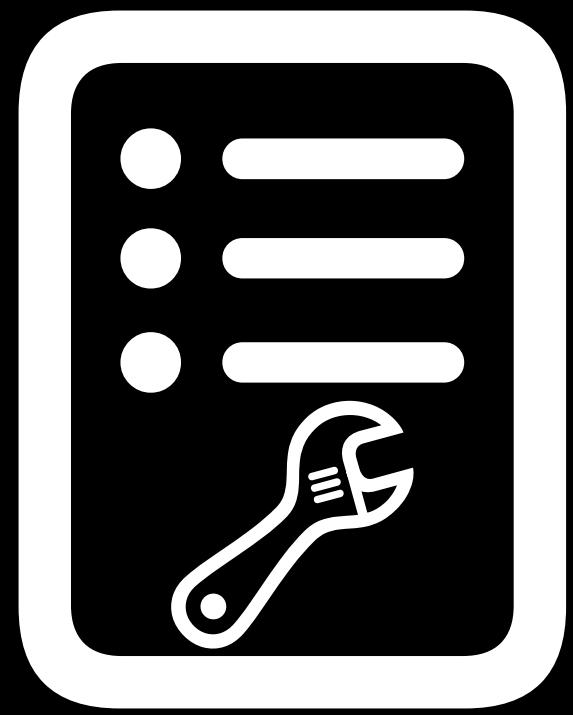
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...>
<plist version="1.0">
<dict>
    <key>Description</key>
    <string>Downloads the latest version of GitHub Desktop.
```

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

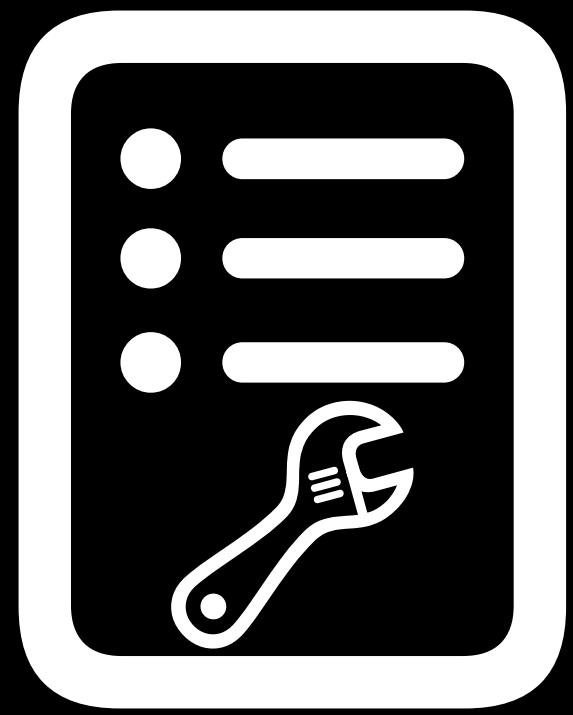
- Can make/run multiple overrides for same recipe
 - ▶ Example: Non-Universal: One Intel, one Apple Silicon

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_x86_64</string>
    <key>Input</key>
    <dict>
        <key>BUILD</key>
        <string>darwin</string>
```

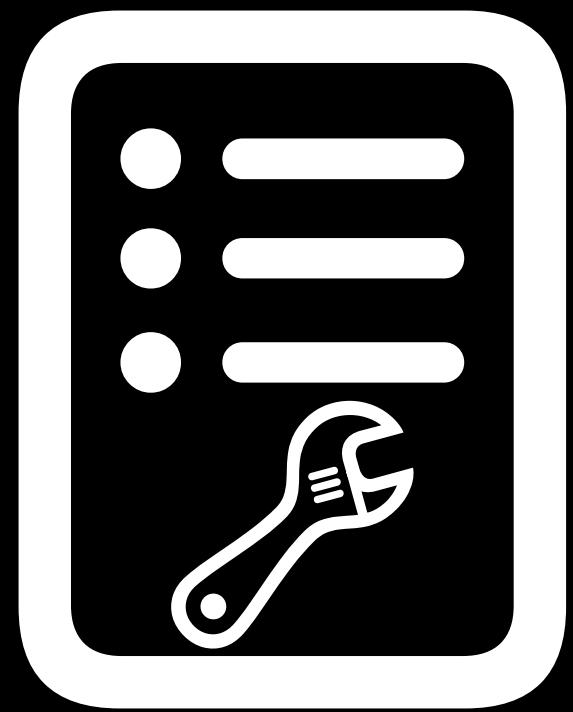
GitHubDesktop_arm64.munki.recipe
(override)

```
<?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>
```



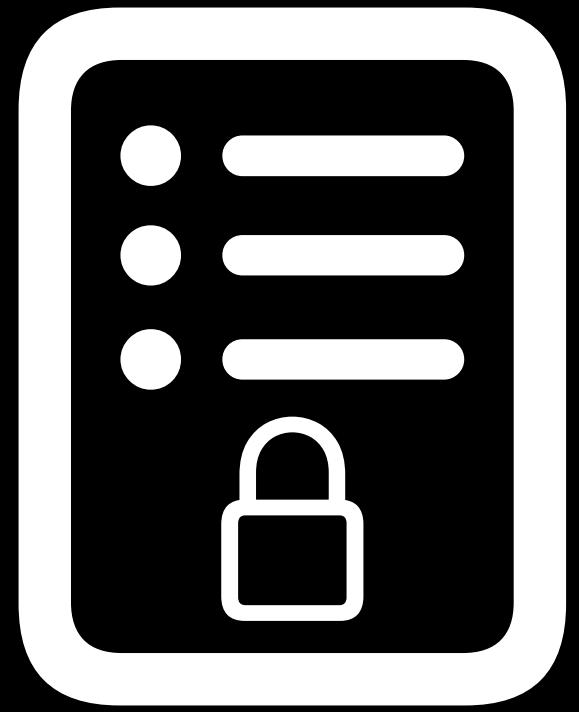
Variables in Overrides

- 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



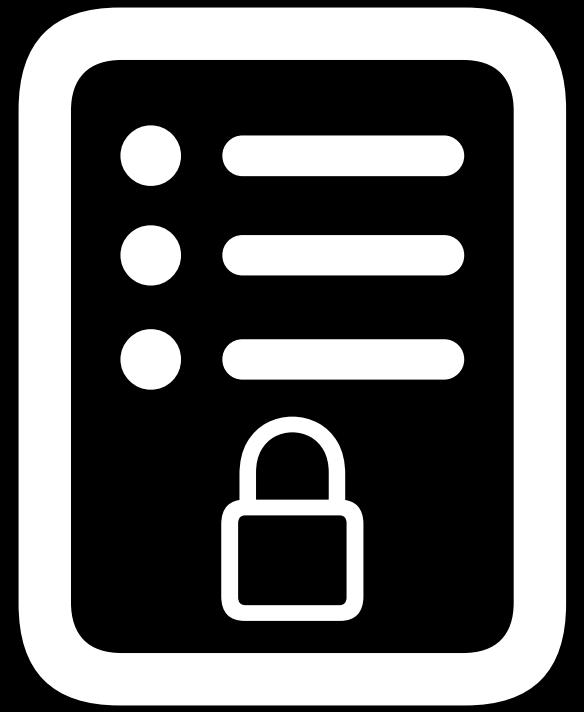
Variables in Overrides

- 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

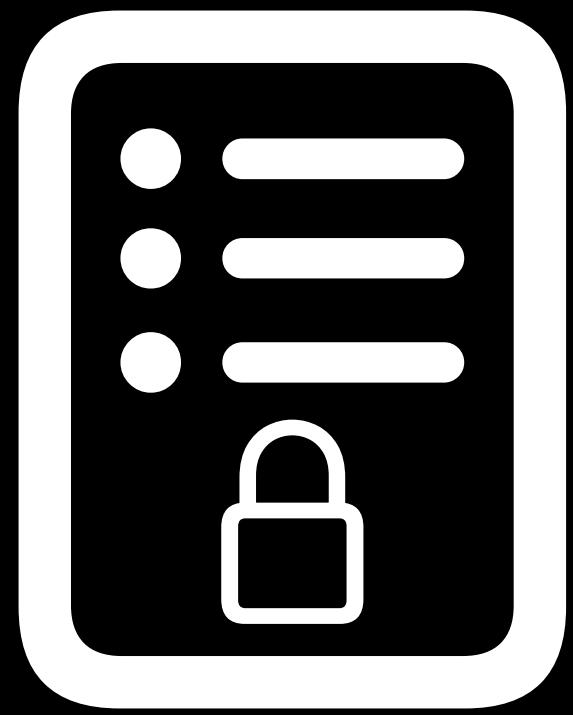
- Allows you to audit changes
- Pins your process to a known state
- Canary indicator for parent updates



Trust Info in Overrides

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

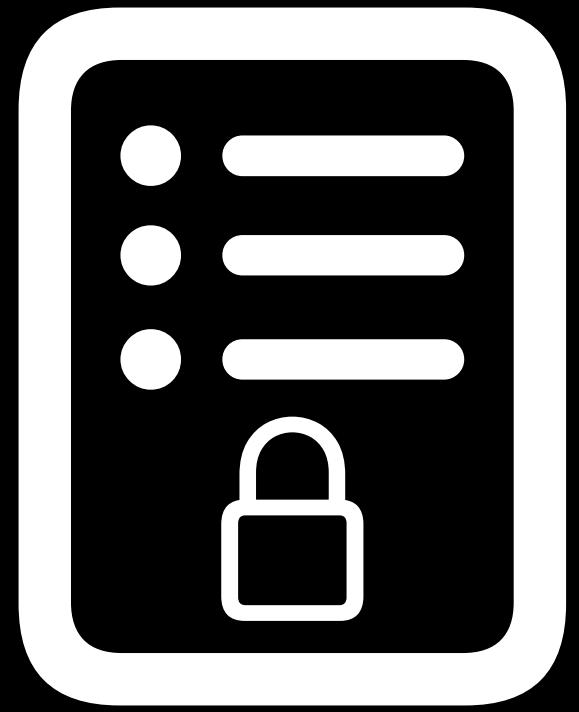
```
defaults write com.github.autopkg FAIL_RECIPES_WITHOUT_TRUST_INFO -bool YES
```



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

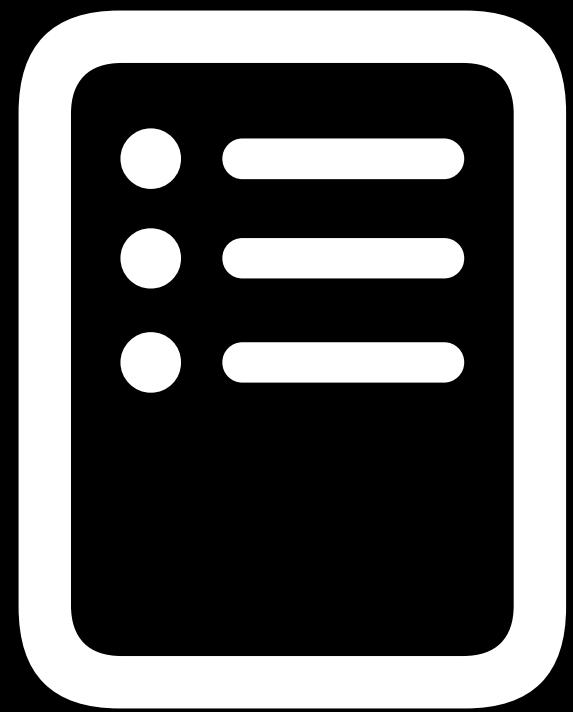
Testing your recipes

Sharing your recipes

Advanced topics (you tell us!)

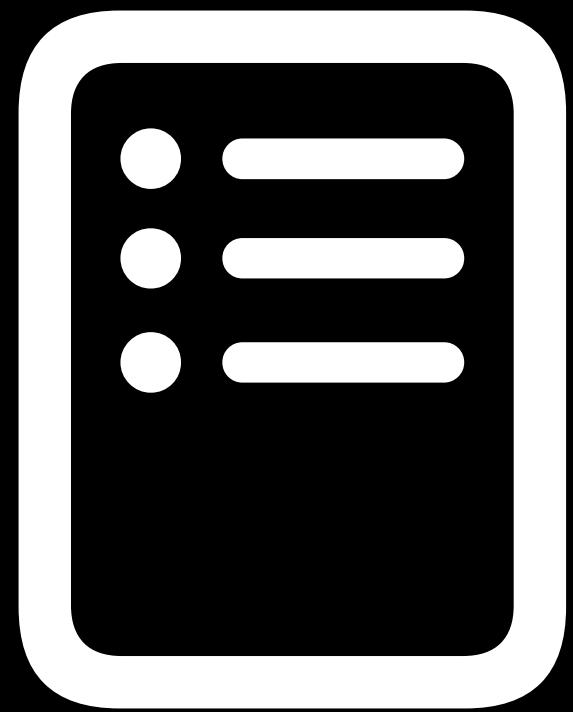
Wrap up

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



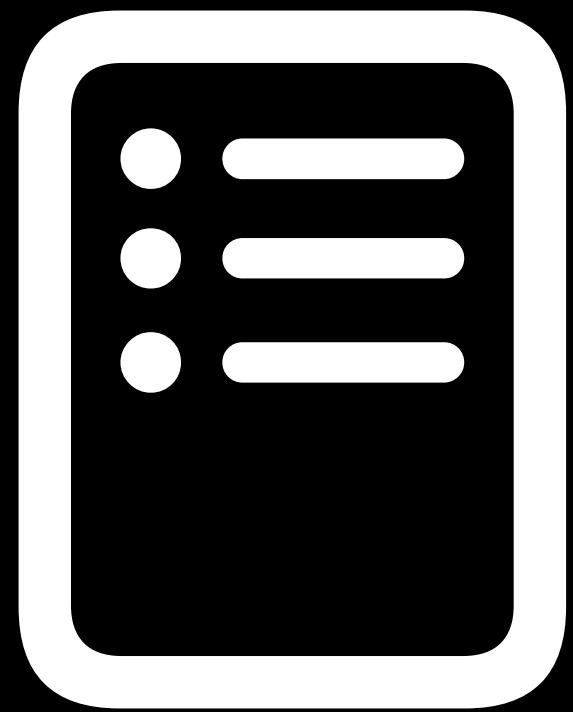
Recipe Lists

- 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.*



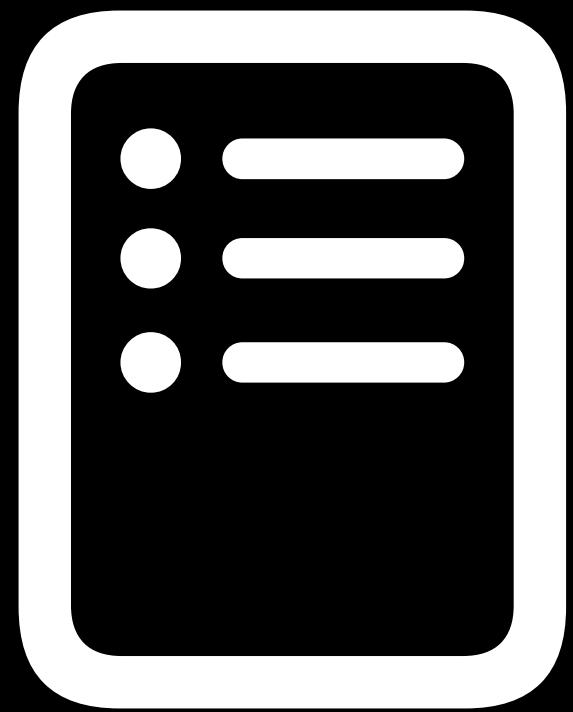
Recipe Lists

- 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



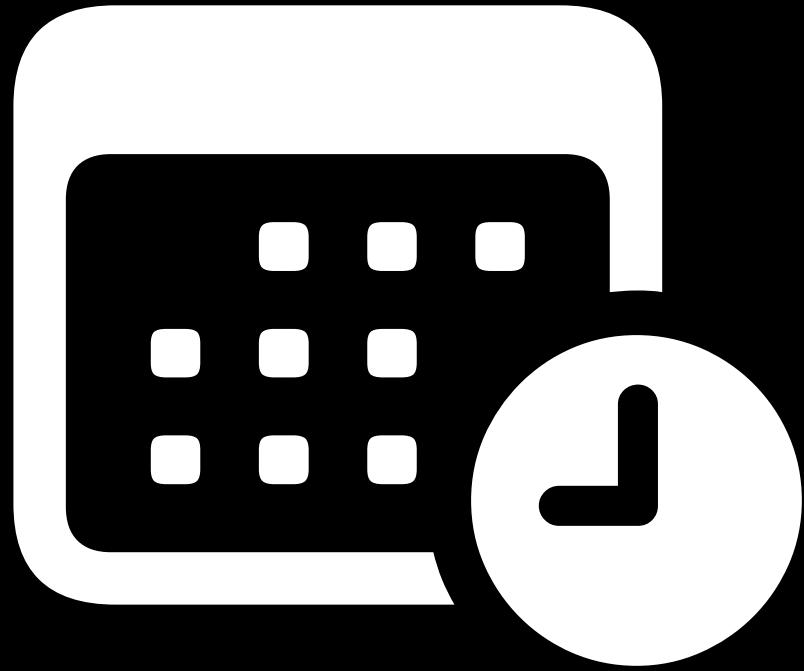
Recipe Lists

- 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



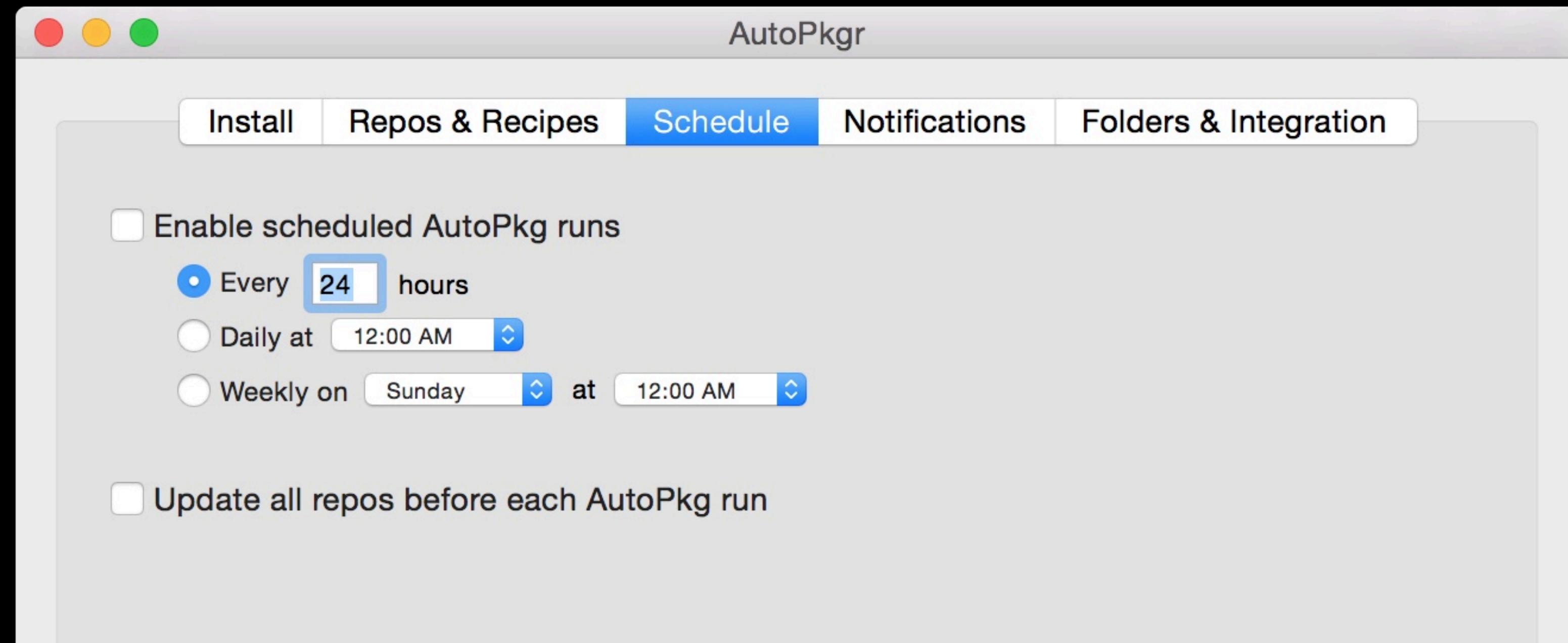
Recipe Lists

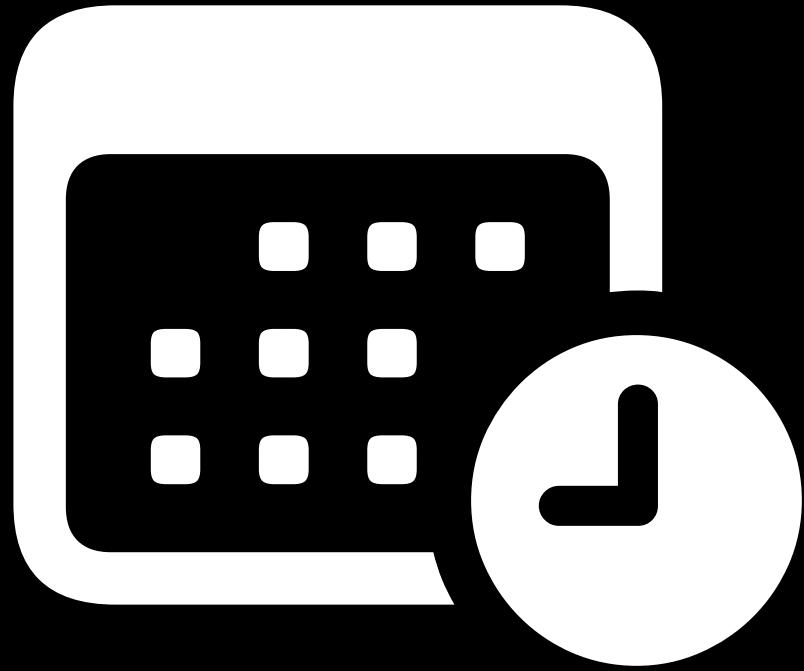
- verify-trust-info verb supports lists
- update-trust-info verb does NOT support lists (by design)
 - ▶ *Does support wildcards, but use discretion*
- Tip from Anthony:
 - ▶ *Verify recipe list, then follow up with -vv on failed recipes*



Scheduling AutoPkg

- AutoPkgr
 - ▶ <https://github.com/lindegroup/autopkgr/wiki/Scheduling-and-Notifications>

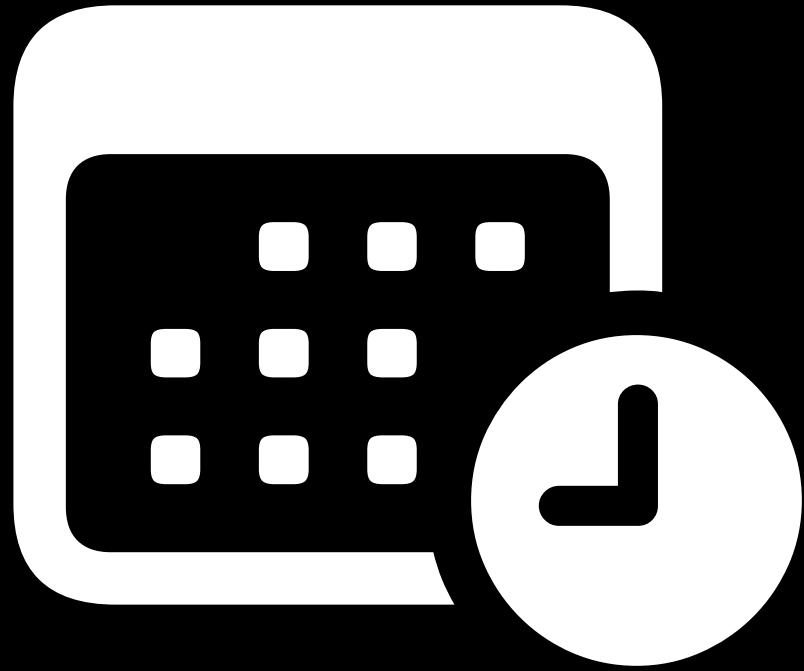




Scheduling AutoPkg

- LaunchAgent and runner script

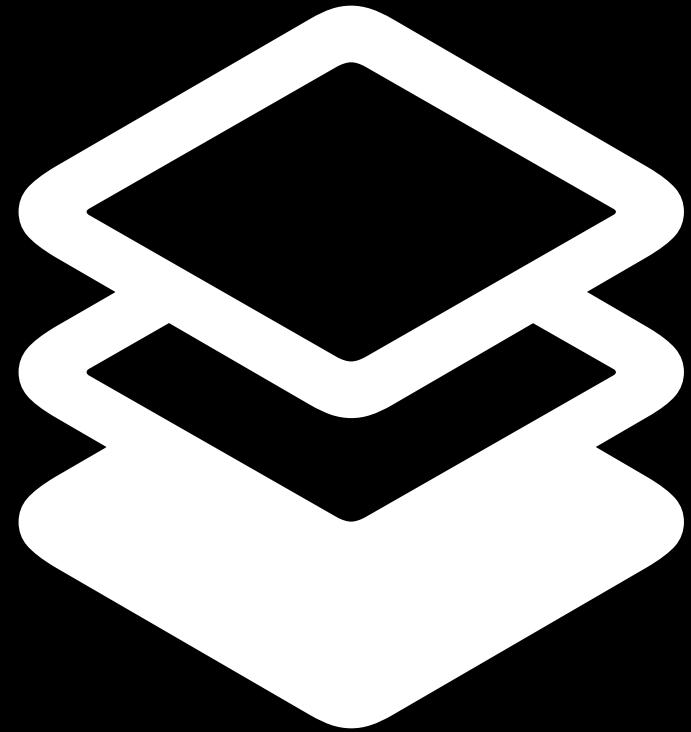
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.betterbag.cpe.autopkgrunner</string>
    <key>ProgramArguments</key>
    <array>
        <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

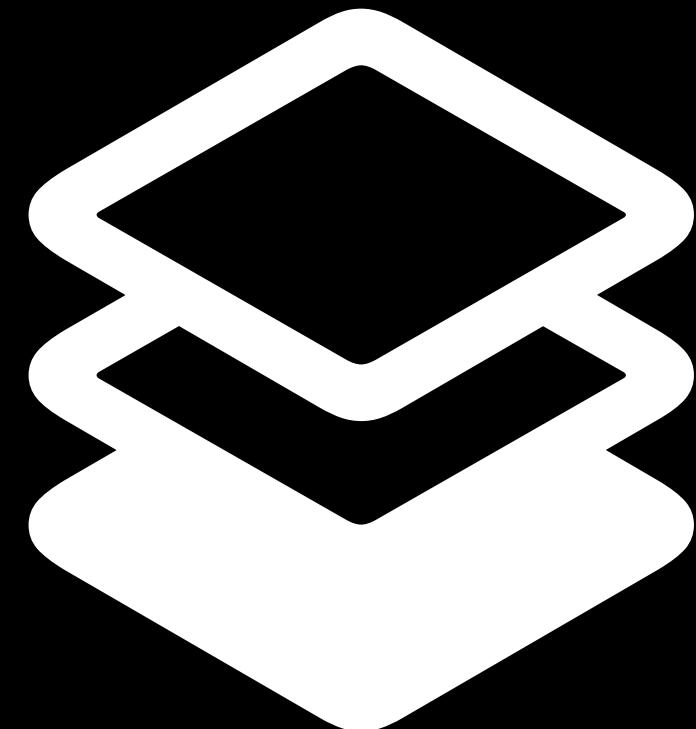
GoogleChrome.ds.recipe

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

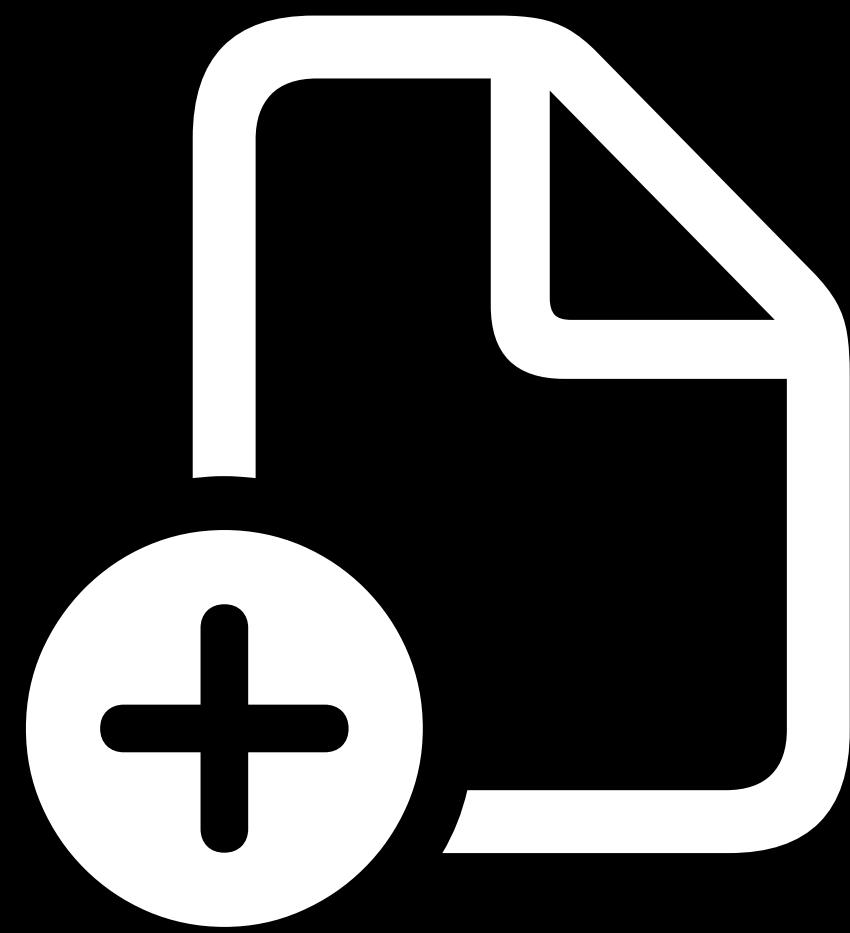
Sharing your recipes

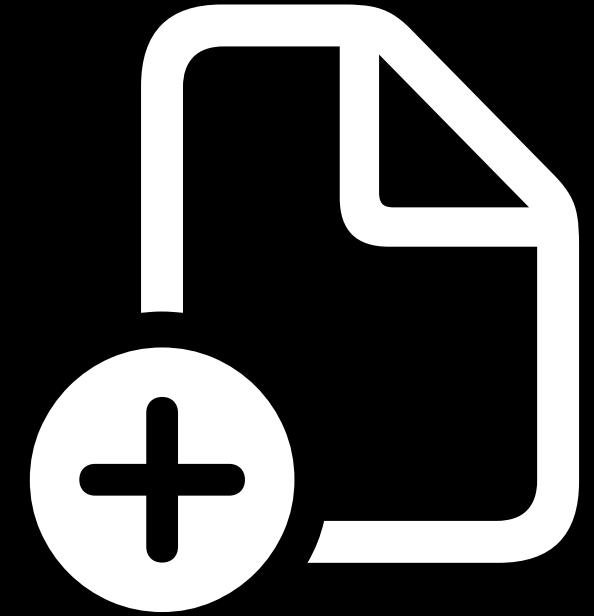
Advanced topics (you tell us!)

Wrap up

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

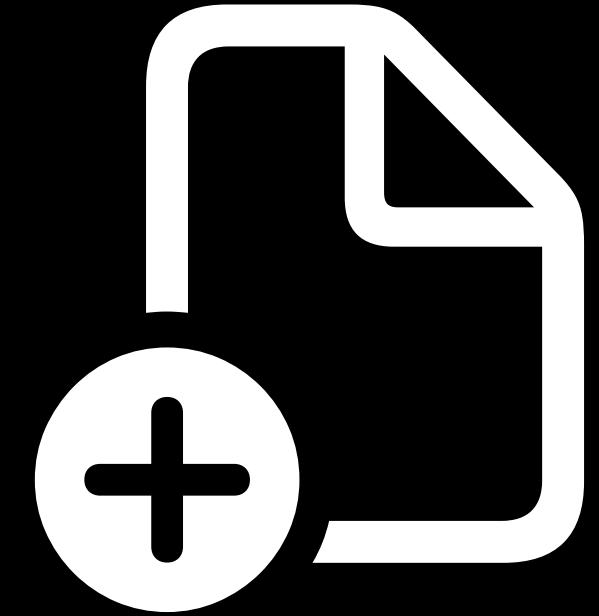
Creating New Recipes





First: Search Thoroughly

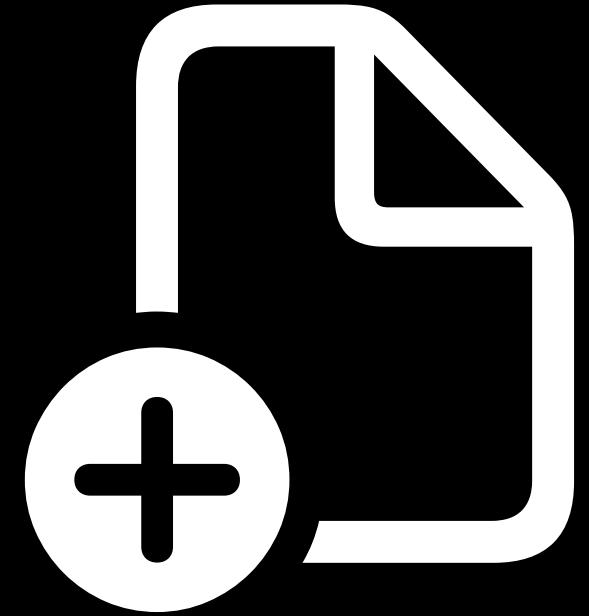
- AutoPkg search verb
 - ▶ As mentioned before, not 100% reliable as of now
- GitHub web search
 - ▶ [https://github.com/search?
type=code&q=org%3Aautopkg+thunderbird](https://github.com/search?type=code&q=org%3Aautopkg+thunderbird)



Creating a Recipe

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





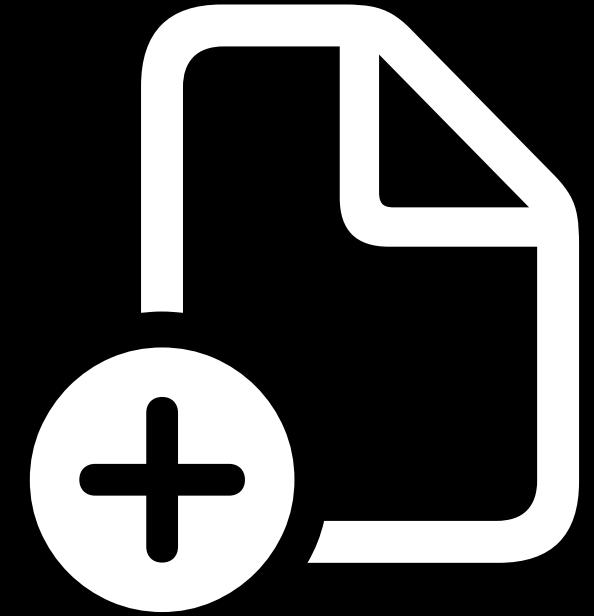
Creating a Recipe

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



`autopkg new-recipe recipename.function.recipe`

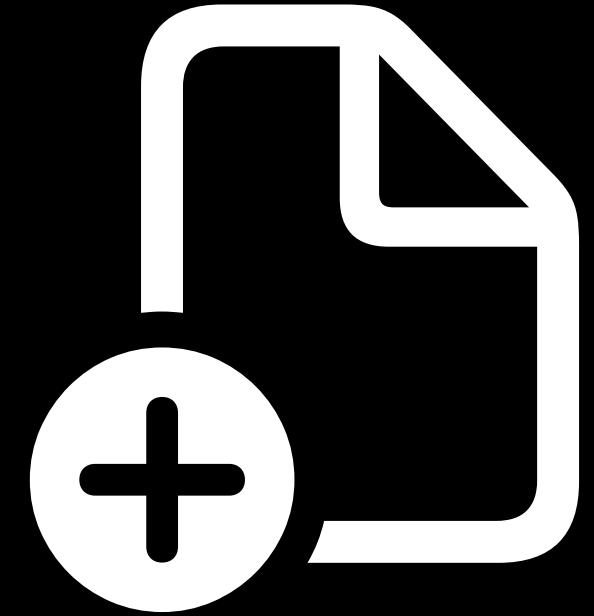
- Leverage an existing parent recipe if you can



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>
```



Creating a Recipe

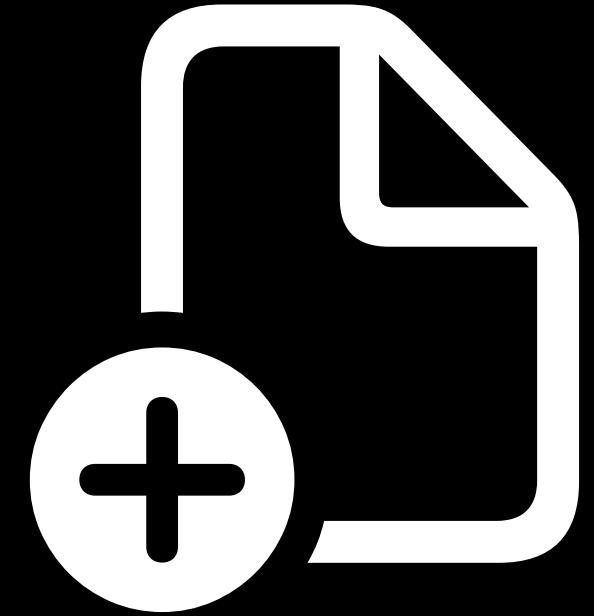
- Many recipes share same DNA
 - ▶ *Static download URL, zipped app, unsigned:*
 - `URLDownloader`, `EndOfCheckPhase`
 - ▶ *Sparkle feed, dmg app, signed*
 - `SparkleUpdateInfoProvider`, `URLDownloader`, `EndOfCheckPhase`, `CodeSignatureVerifier`



Recipe Robot

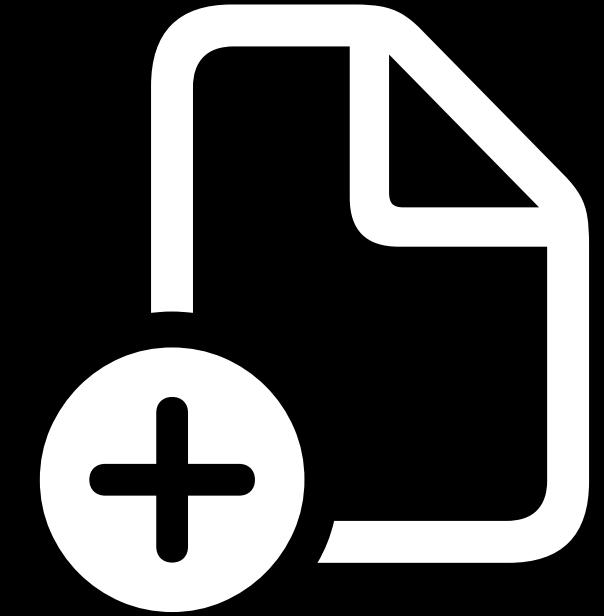
<https://github.com/homebysix/recipe-robot>

- 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



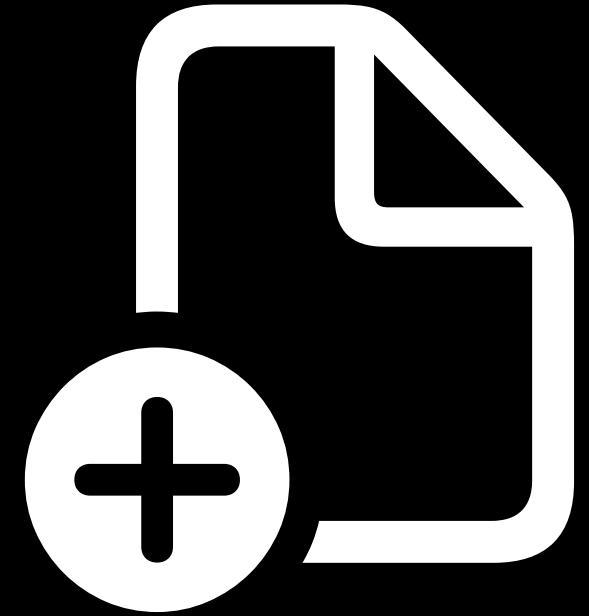
New-Recipe Verb

- 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



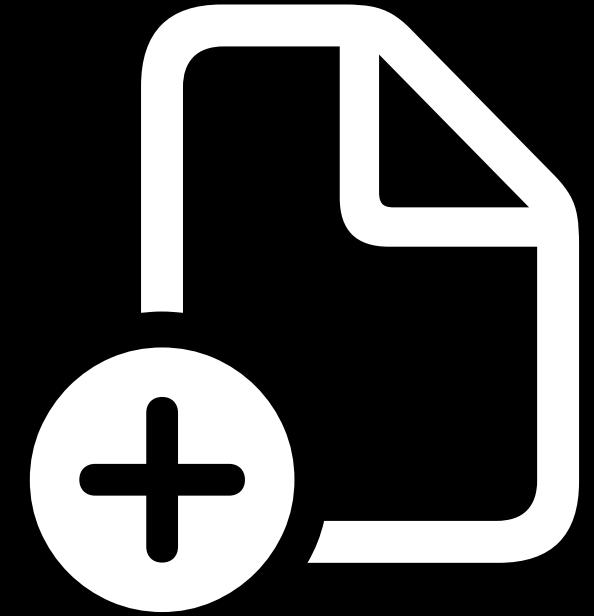
Retrieve Code Signature

- 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

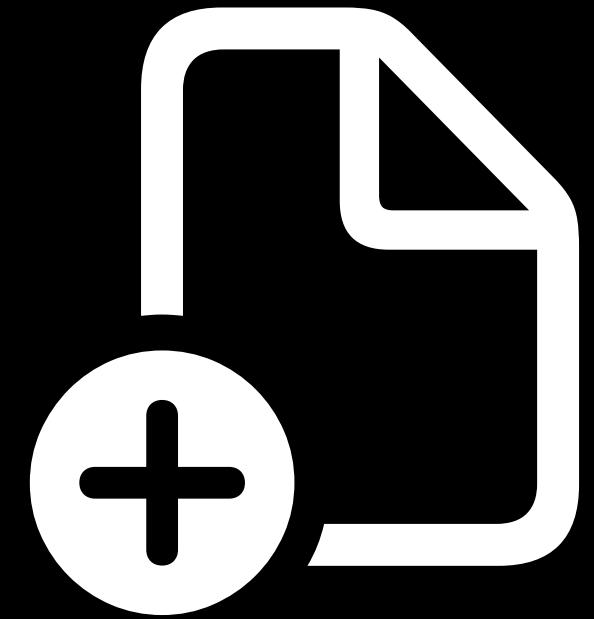
Input Variables

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

Output Variable

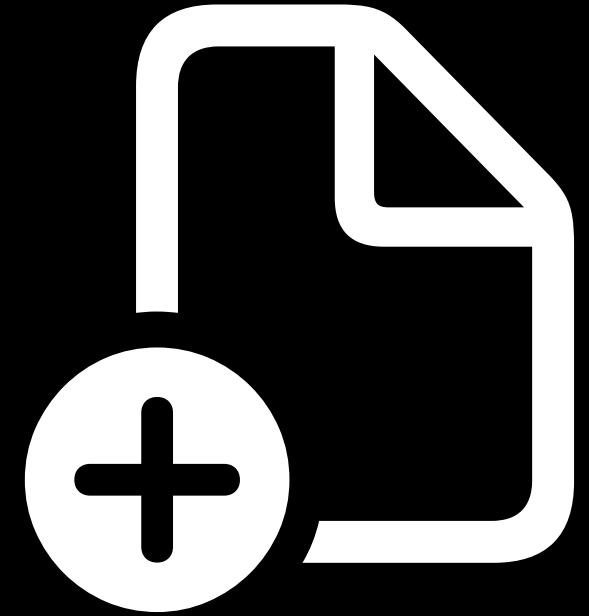
- `result_output_var_name`

* required



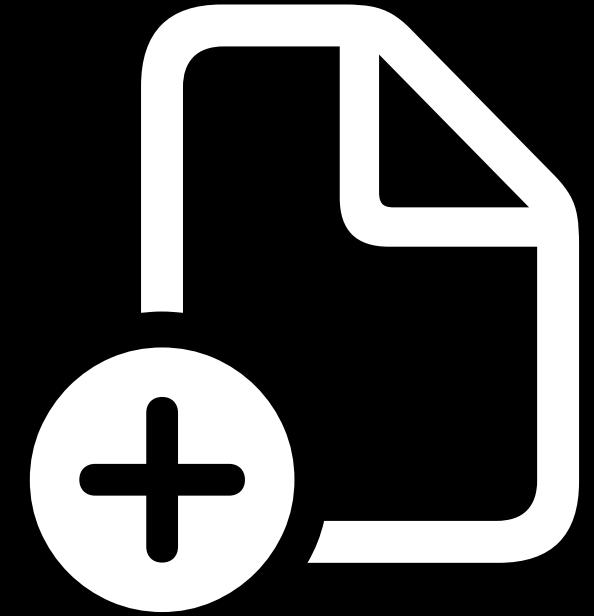
Start the Search

- 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
 - Session Video: youtu.be/Wc8Kpw0nEww
- Other options: [Patterns.app](https://patterns.app), regex101.com

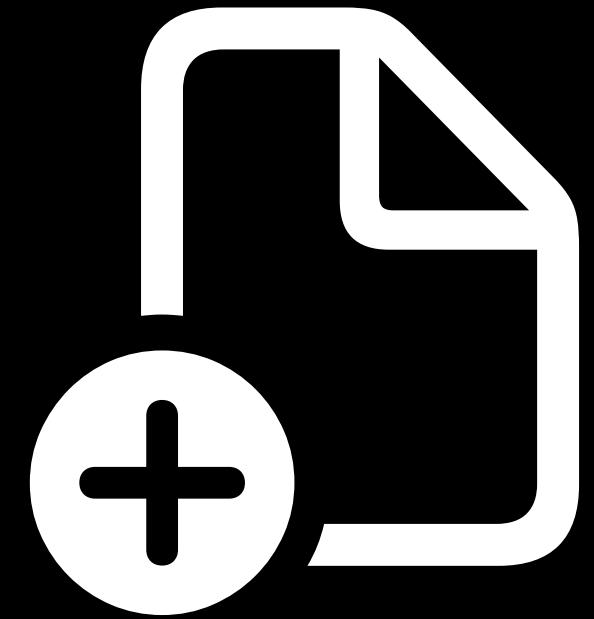


RegEx

- Test in your recipe
 - ▶ *Plist: encode < and > (< >)*
 - ▶ *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/



Recipe Editing Pro Tips

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

VSCode example settings.json
file section:

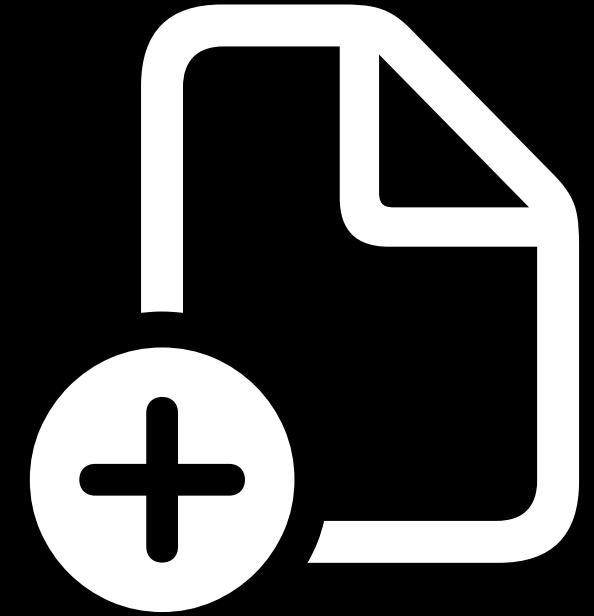
```
"files.associations": {  
    "*.plist": "xml",  
    "*.recipe": "xml",  
    "*.mobileconfig": "xml",  
    "*.entitlements": "xml"  
},
```

BBEdit example Languages setting:

Custom filename extension mappings:
Use these settings to adjust how BBEdit determines a file's language based on its name.

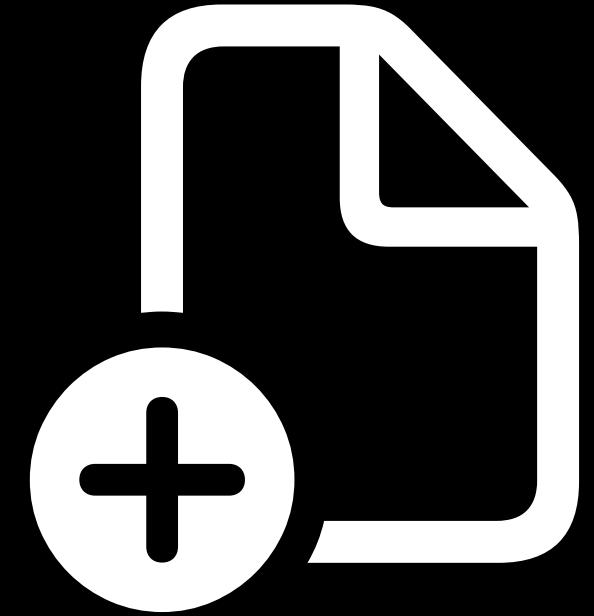
Extension	Language
.recipe	XML

[+] [-]



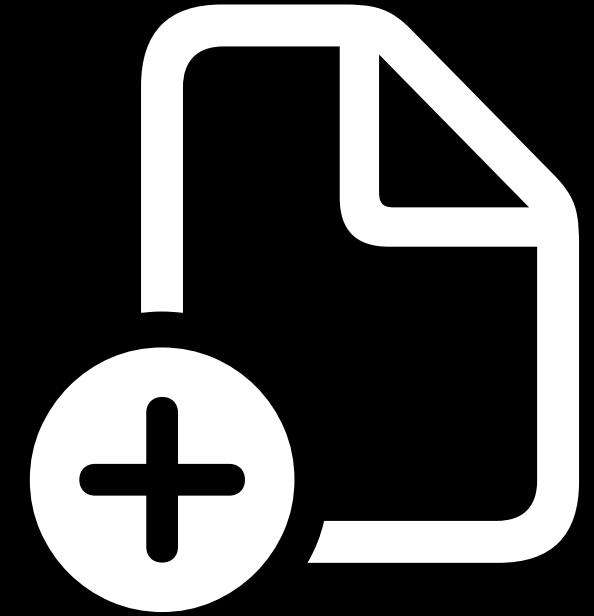
Recipe Editing Pro Tips

- 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*



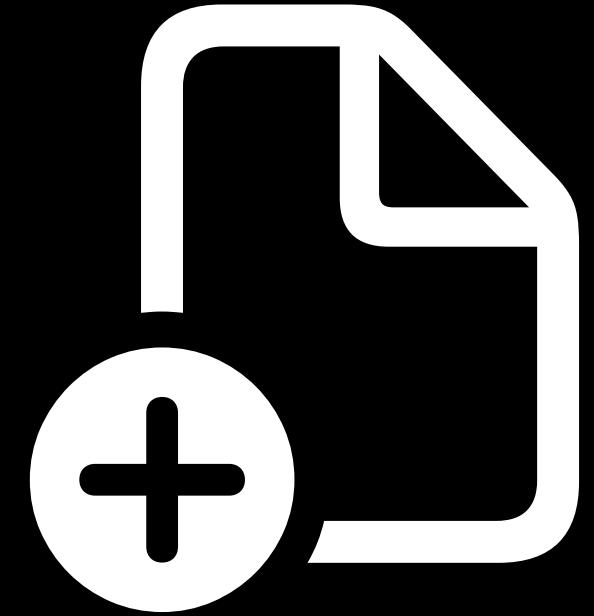
Recipe Editing Pro Tips

- Reusable snippets to save time
 - ▶ *BBEdit: github.com/jazzace/BBEdit-AutoPkg-Clippings*
 - ▶ *TextExpander, Typelt4Me, Typinator, Keyboard Maestro, Alfred*



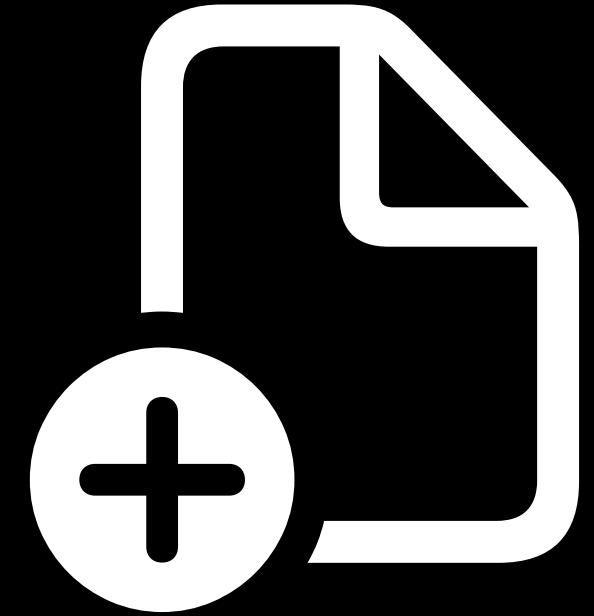
Recipe Editing Pro Tips

- 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*



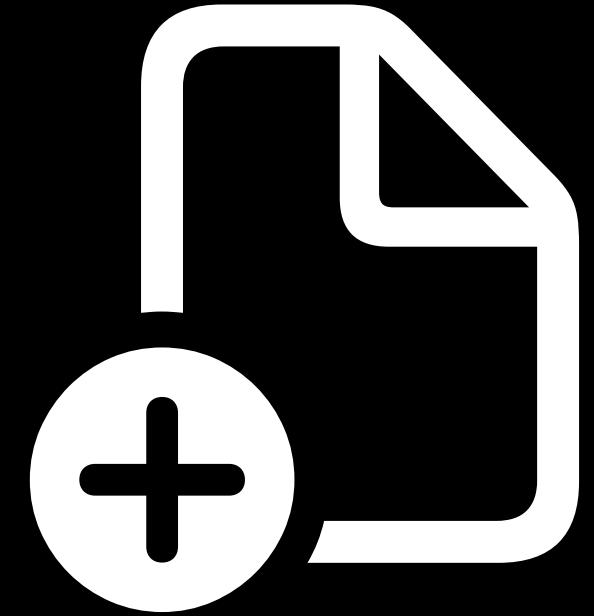
Recipe Editing Pro Tips

- 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



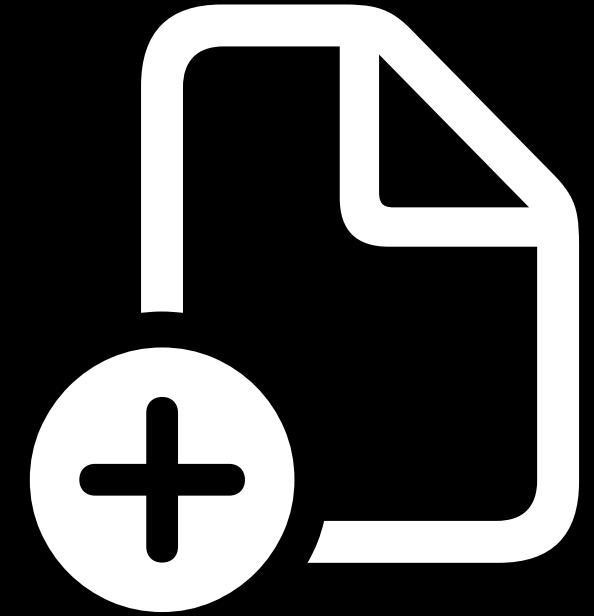
Recipe Editing Pro Tips

- 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"



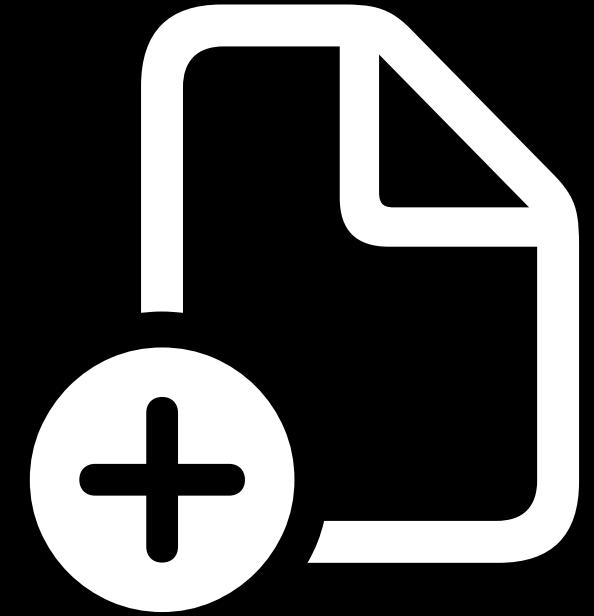
Recipe Editing Pro Tips

- 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

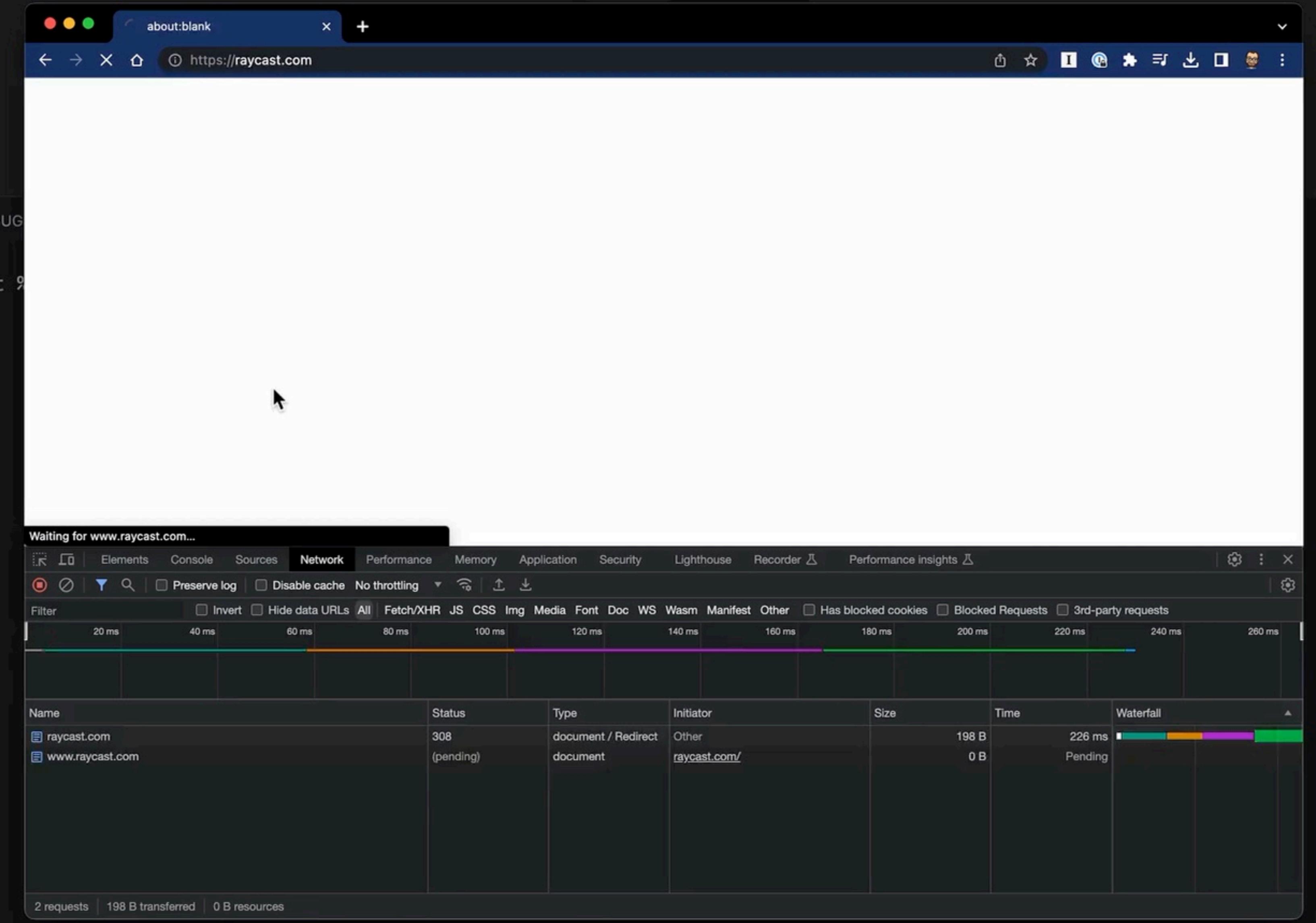
- Good to keep as a snippet
- If download URL is dynamic, you can:
 1. *Provide it to Recipe Robot to get the recipes created*
 2. *Edit the download recipe to feed URLDownloader with URLTextSearcher*



DEMO

- Building a recipe with Recipe Robot



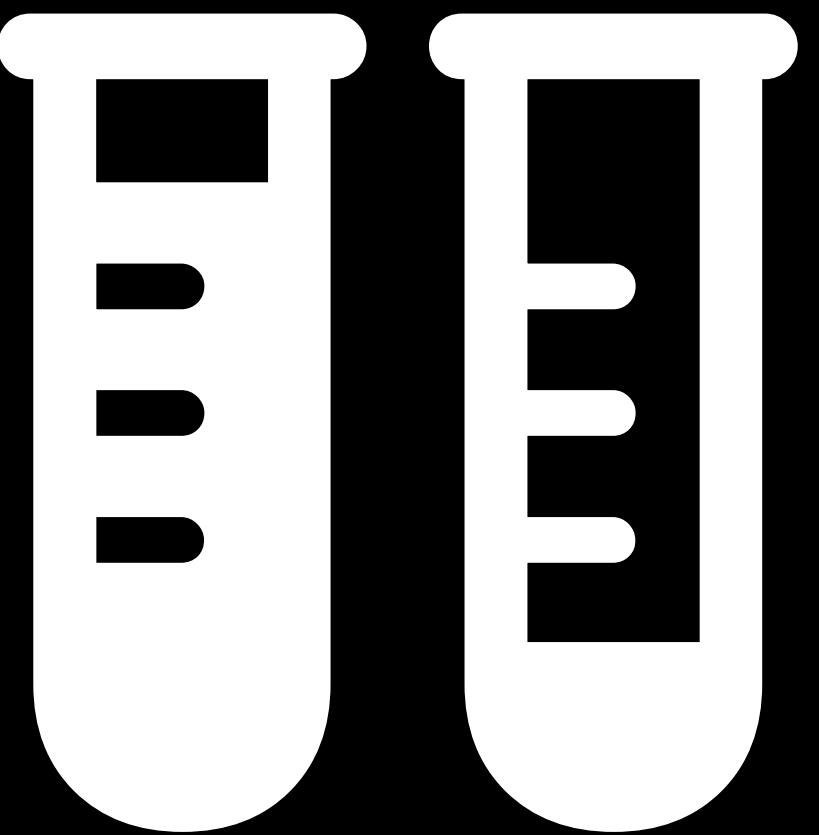


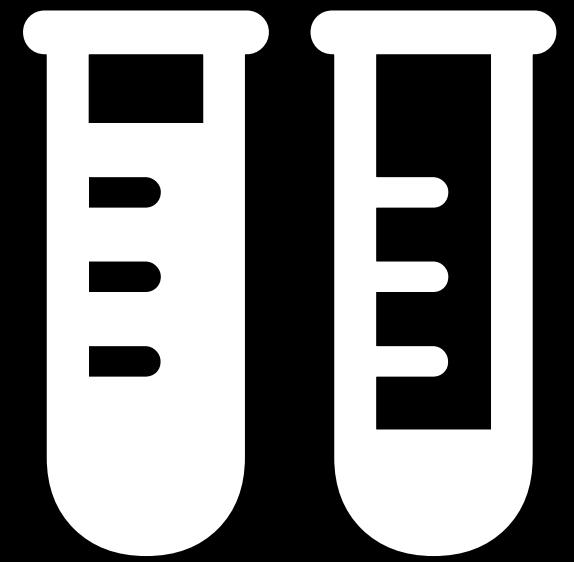
Your Turn!



- 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
- 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

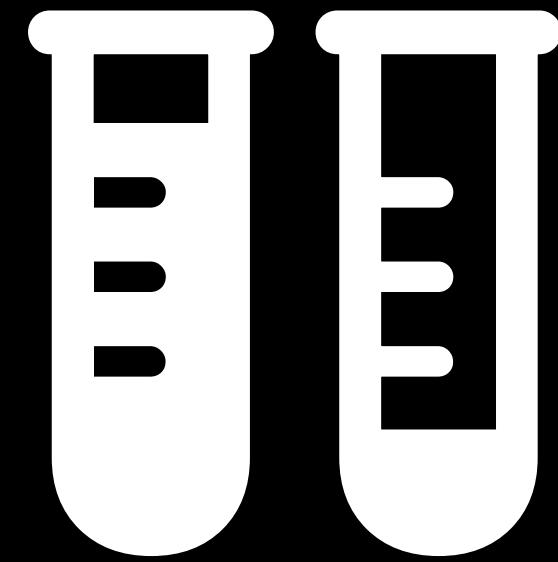
Testing Your Recipes





Recipe Search Order

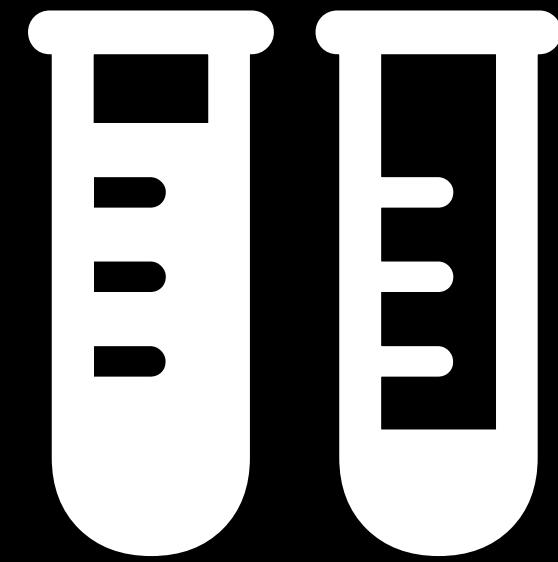
```
% defaults read com.github.autopkg RECIPE_SEARCH_DIRS
(
    ".",
    "~/Library/AutoPkg/Recipes",
    "/Library/AutoPkg/Recipes",
    "/Users/areimer/Library/AutoPkg/RecipeRepos/
com.github.autopkg.recipes",
    "/Users/areimer/Library/AutoPkg/RecipeRepos/
com.github.autopkg.jazzace-recipes",
  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



Testing 1, 2, 3

(With or without overrides)

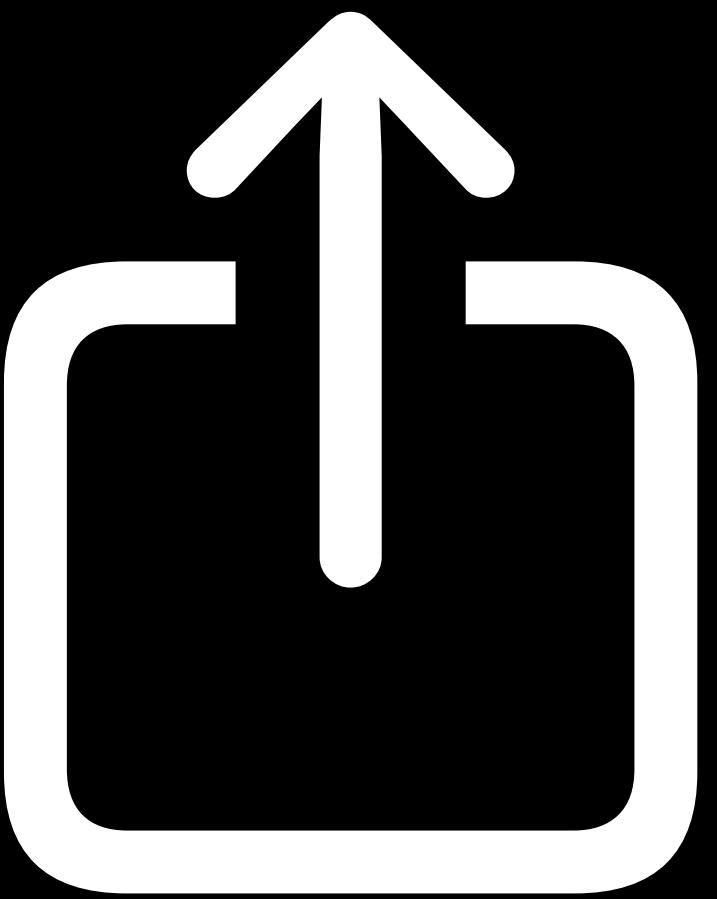
1. Test the parent(s)
2. Test my new child recipe
3. Test it again

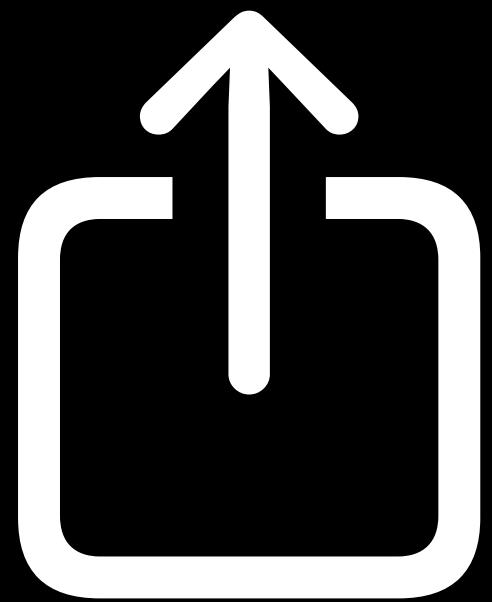


Testing Tips

- ▶ `autopkg run recipename.type --pkg /path/to/download`
- ▶ `alias apr='autopkg run -vv'`
- ▶ `alias apt='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





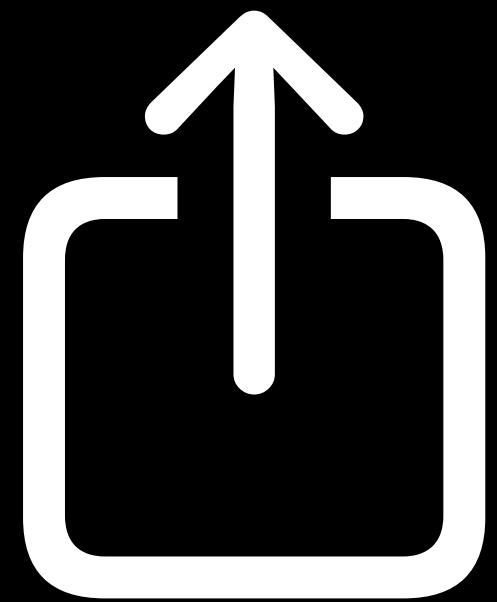
Sharing Your Recipes

- Why?

- ▶ *Appear in search results*
- ▶ *Saves community time*
- ▶ *Establishes you as recipe author*

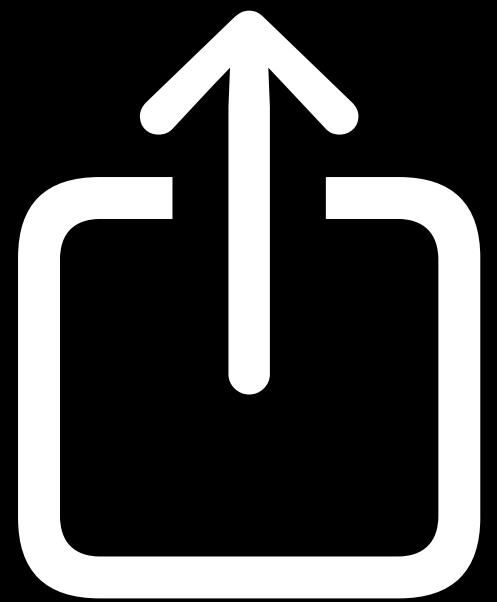
- Why not?

- ▶ *Internal to your company*
- ▶ *Download behind paywall*
- ▶ *You don't have time/interest in maintenance*



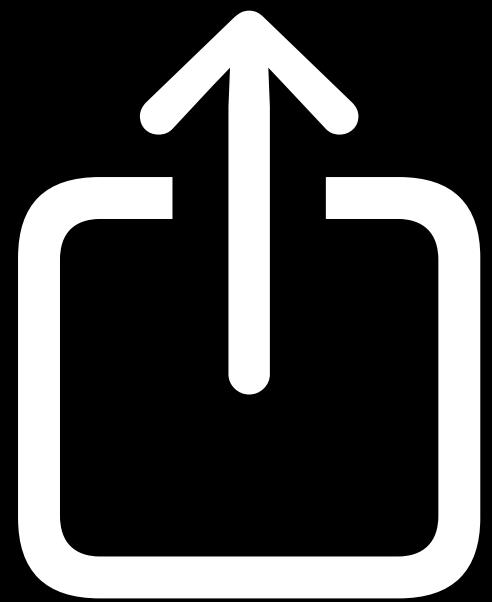
Sharing Your Recipes

- 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



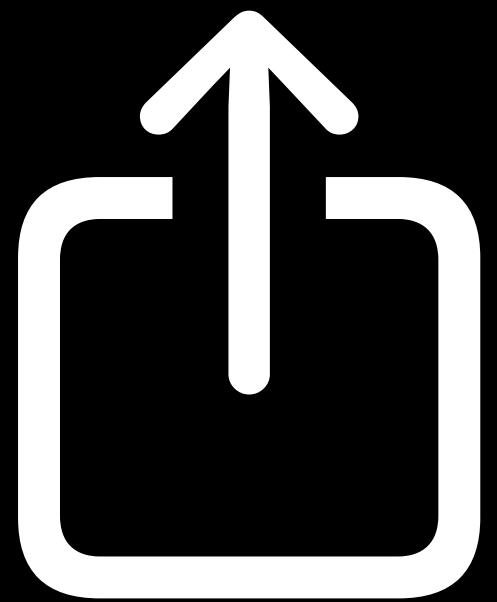
Sharing Your Recipes

- Recipe maintainer expectations
 - ▶ github.com/autopkg/autopkg/wiki/Sharing-Recipes#repo-maintenance-expectations



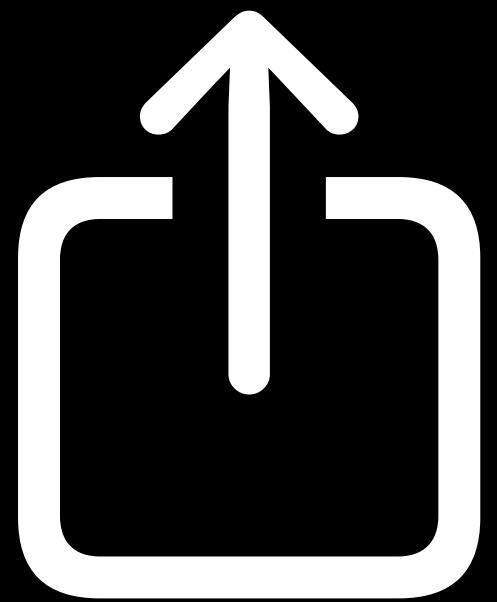
Sharing Your Recipes

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



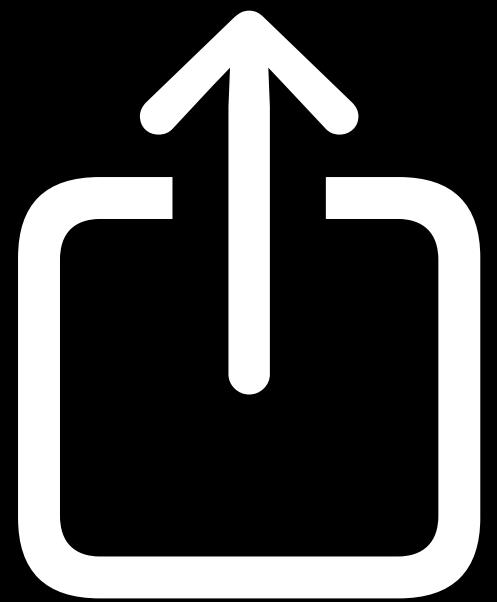
Sharing Your Recipes

- 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*

/ \ 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 -vvvcq "{}" '+'`
 - ▶ *Probably OK to skip Munki, Jamf, and Install recipes*
- Getting help
- Recipe deprecation/ownership transfer
 - ▶ *Tip: DeprecationWarning and change parent to replacement recipe*

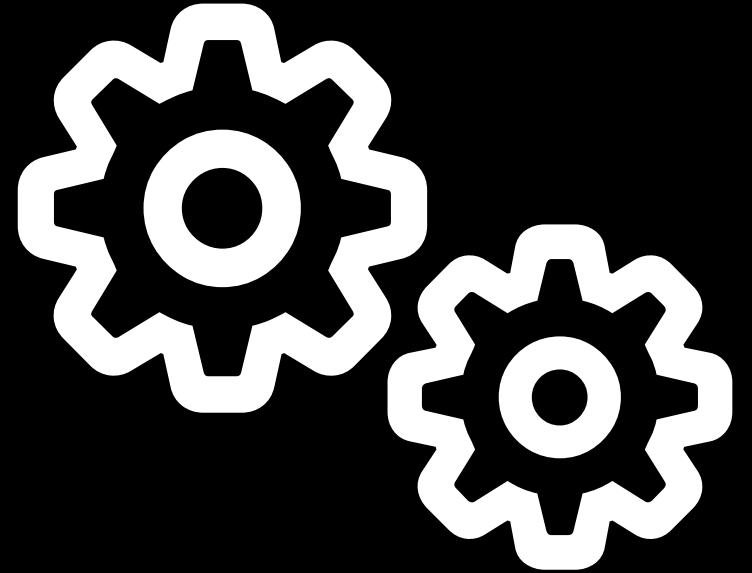
Outline

- ✓ 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!)

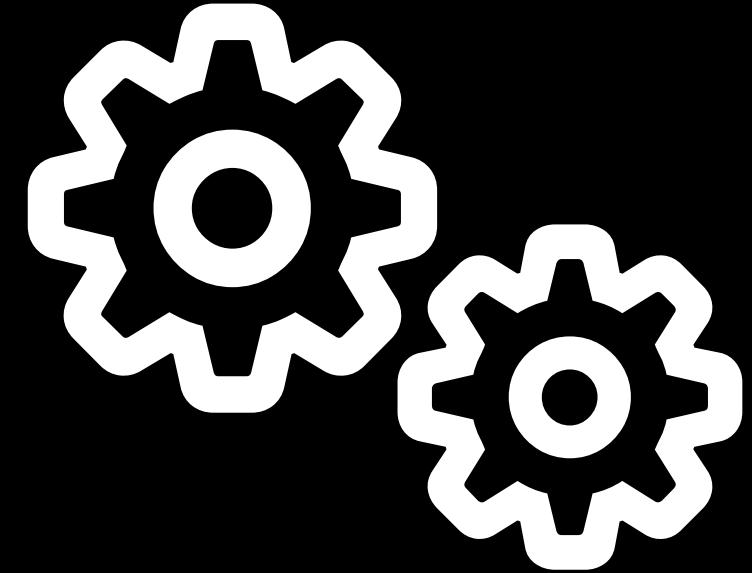
Wrap up

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



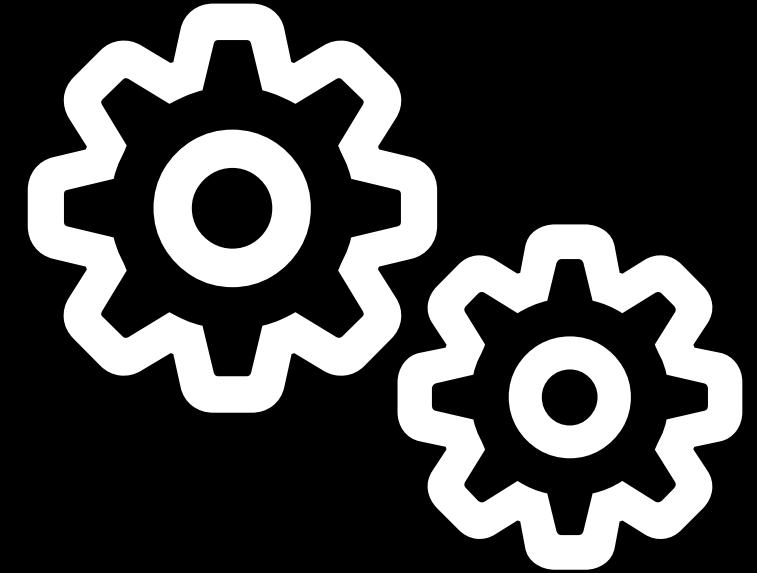
Processor Anatomy

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



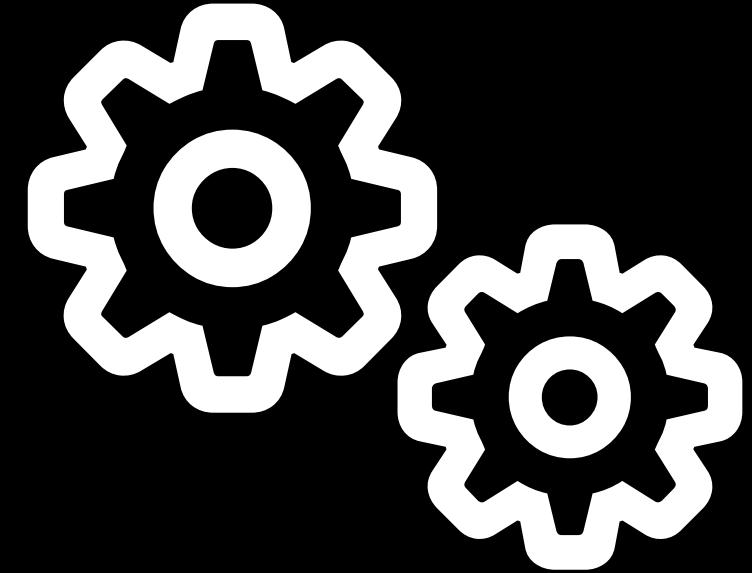
Processors as Templates

- 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

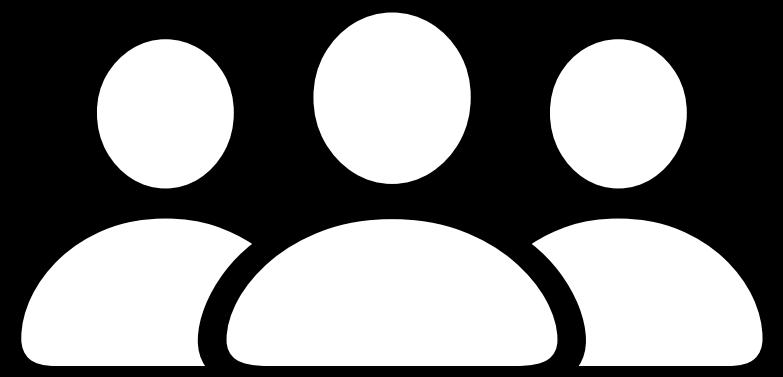
- Broadly useful and flexible
 - ▶ *VendorURLProvider instead of ProductURLProvider*
- Don't assume English
 - ▶ *Add LOCALE input variable if multiple are available*
- Provide useful output variables
- Ensure Python 3 compatibility (RIP Python 2)



Pro Tip: URLGetter

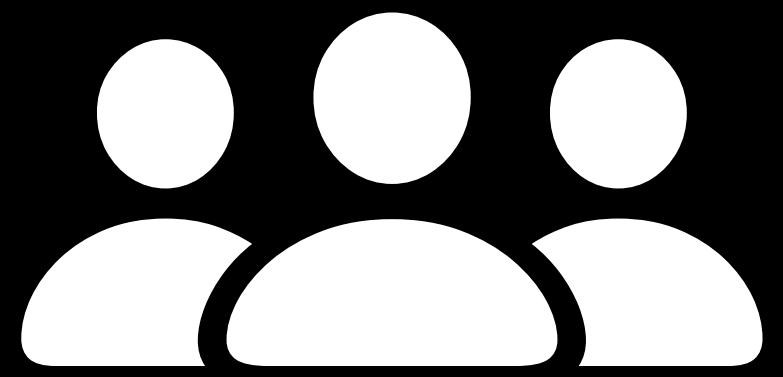
- Superclass used by processors that download things
- Consider using URLGetter instead of importing urllib or subprocessing curl

Contributing to AutoPkg



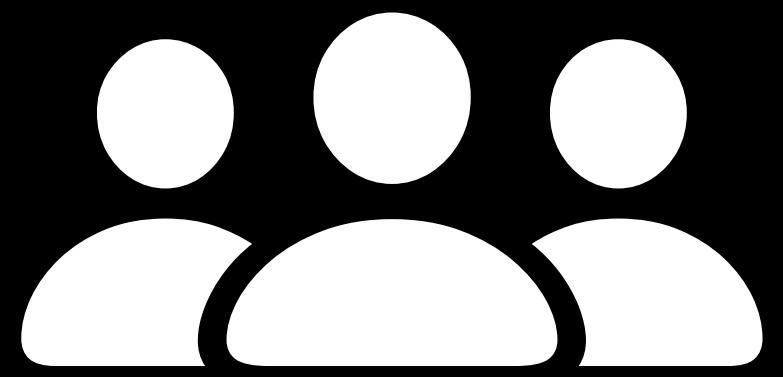
AutoPkg Community

- 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 f 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

nmfspadden 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 ...

Reviewers

No review

Assignees

No one as

Labels

Alternative syntax for repo-list and list-processors

 Add verbs list-repos and processor-list 

0ee0753

Conversation 2

Commits 1

Checks 0

Files changed 1



jazzace commented on Apr 21

Member

...

Reviewers

No review

Assignees

No one as

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked iss

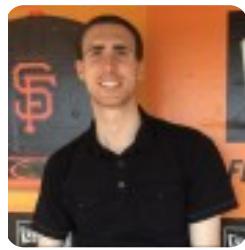
Alternative syntax for repo-list and list-processors



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:

[Code](#)[Issues 19](#)[Pull requests 3](#)[Actions](#)[Wiki](#)[Security 0](#)[Insights](#)

Issue: Bug report

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



Title

[Write](#) [Preview](#)

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)

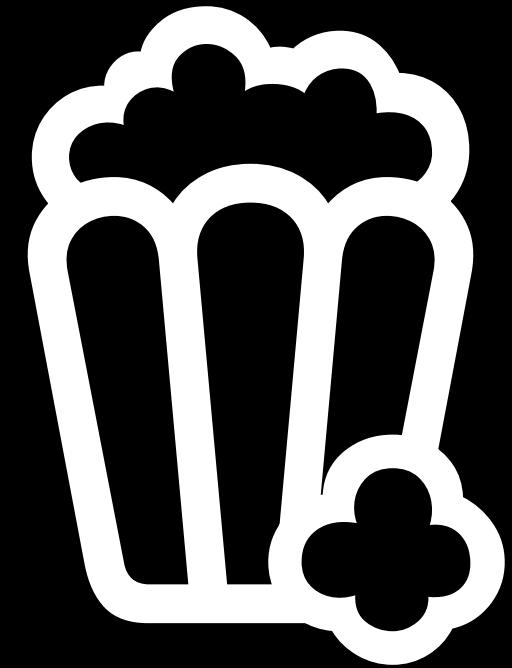
Styling with Markdown is supported

[Submit new issue](#)

Helpful resources

[Contributing](#)

[GitHub Community Guidelines](#)



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>*
- Rich Trouton (**rtrouton**)
 - ▶ *<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>

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

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

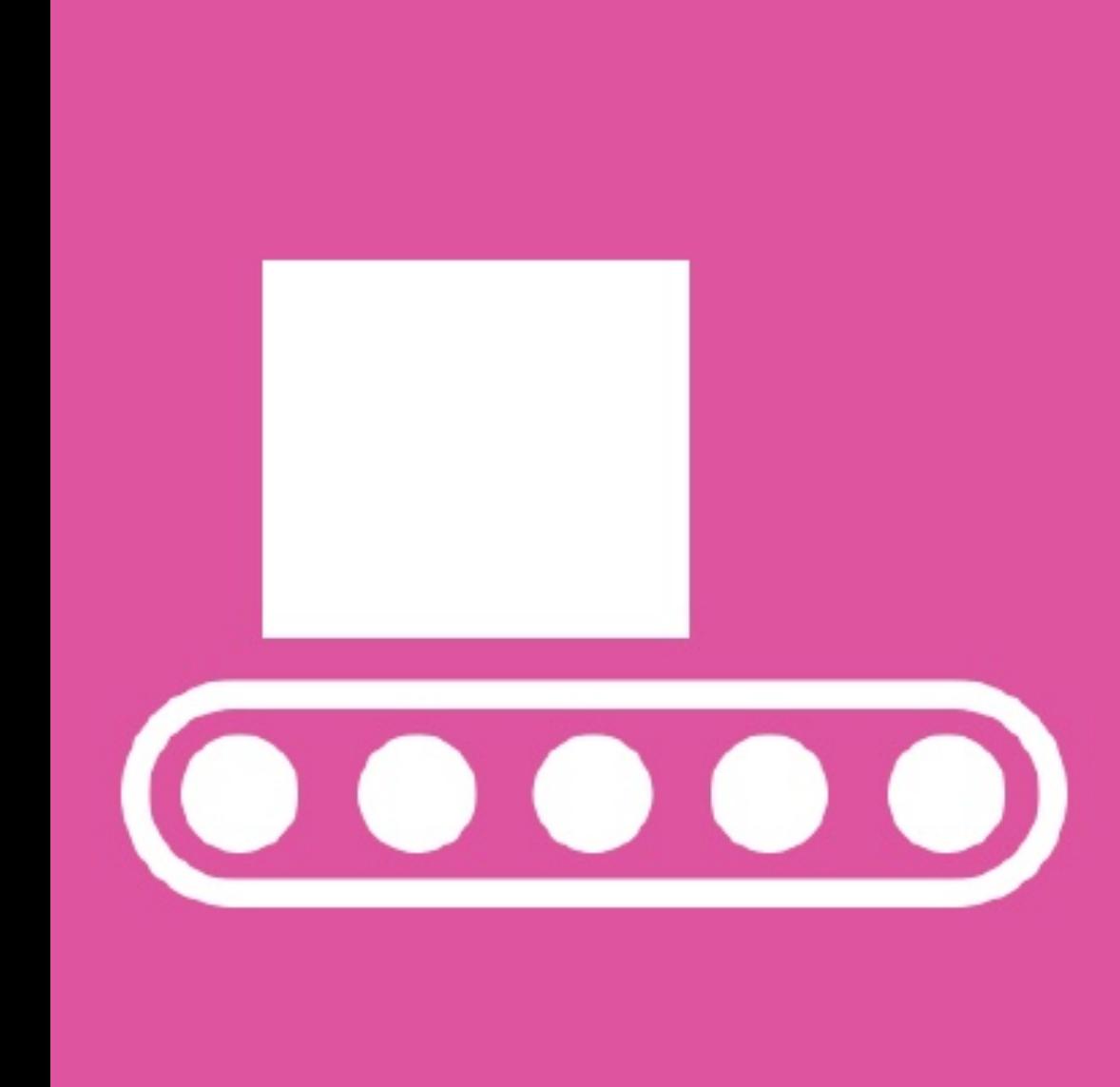
Links and References: <https://tinyurl.com/AutoPkgWorkshop>

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>