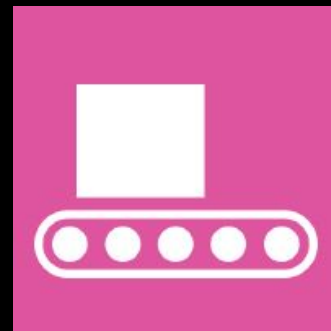


AutoPkg for Windows Software 2.0

James Stewart
@JGStew

<https://github.com/jgstew/MacAdmins2023>



Links to Links:

<https://github.com/jgstew/MacAdmins2023>



BigFix

I work for BigFix

I am NOT here to talk about BigFix

My example is BigFix

Agenda

- Why AutoPkg?
- What is AutoPkg? Recipes and Processors
- AutoPkg for Windows Software 1.0
- URLDownloader Cross Platform Problems
- AutoPkg for Windows Software 2.0
- URLDownloaderPython Processor
- More Processors
- Project Results
- Project Challenges
- Solutions
- Examples and Demos

?

¡Please ask Questions anytime!

?

¿ Who has heard of AutoPkg before ?

?

¿ Who has used AutoPkg before ?

?

¿ Who attended "AutoPkg in Depth" ?

with Elliot and Anthony

?

¿ Who attended "Bringing a Mac Admin Mindset to Managing Windows Endpoints" ?

with Rebecca and Adam

?

¿ Who does NOT use or manage Windows ?

Why AutoPkg?

Add New Software Version to Deployment Tool

Is there new software?

Download binary

Extract metadata

Validate Signatures or Hashes

Put it together in the format for the tool

Put it in the tool

What is AutoPkg?

Originally for automating MacOS software packaging on MacOS

Checks if new software is available

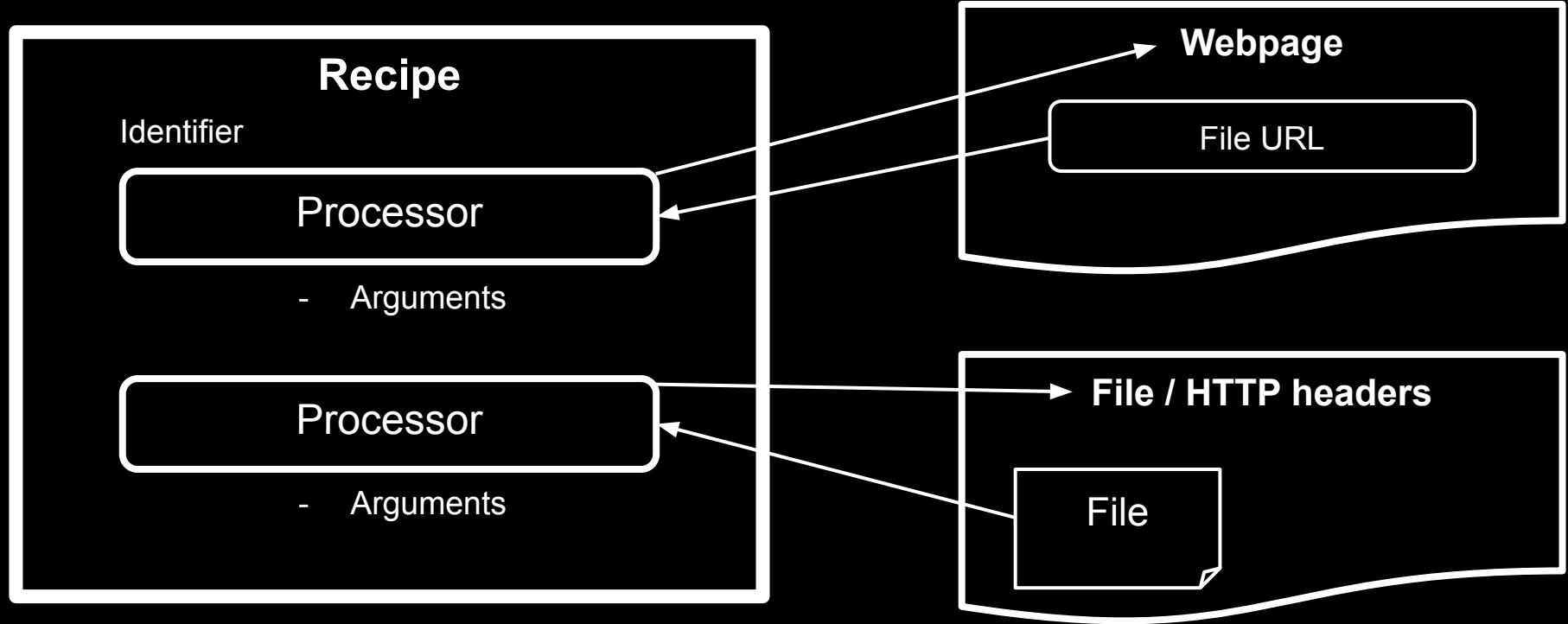
Only downloads software if it has changed

Perform any automated steps required after downloading

Stage software in a software distribution tool (Munki / BigFix / Others)

No "Coding" required for recipes!

What is AutoPkg? Recipes and Processors



What is AutoPkg? Recipes and Processors

Recipes

- Recipes are written in YAML
- Recipes invoke Processors
- Recipes specify arguments
- Recipes can call other Recipes
- Recipes DO NOT contain code

Processors

- Processors are written in Python
- Processors specify input arguments
- Processors specify defaults
- Processors specify outputs
- Processors are the "code"

Recipes do often require regular expressions

Example Recipe: Test Processor Recipe

Description: Test AssertInputContainsString Processor

Identifier: com.github.jgstew.test.AssertInputContainsString

Input:

NAME: AssertInputContainsStringTest

MinimumVersion: "2.3"

Process:

– **Processor:** com.github.jgstew.SharedProcessors/AssertInputContainsString

Arguments:

input_string: "1_2_3"

assert_string: "_"

Example Processor: Inputs and Outputs

```
class AssertInputContainsString(Processor): # pylint: disable=invalid-name
    """checks that assert_string is within input_string"""

    description = __doc__
    input_variables = {
        "input_string": {"required": True, "description": "string to test"},
        "assert_string": {
            "required": True,
            "description": "the string that must be within input_string",
        },
        "raise_error": {
            "required": False,
            "default": True,
            "description": "determines if a failure should raise an error",
        },
    }
    output_variables = {
        "assert_result": {"description": ("The result of the check")},
    }
```


Example Processor: Code

```
def main(self):
    """Execution starts here"""

    input_string = self.env.get("input_string")
    assert_string = self.env.get("assert_string")
    raise_error = bool(self.env.get("raise_error", True))

    self.output(f"`{input_string}` must contain `{assert_string}`", 2)

    try:
        assert assert_string in input_string
        self.env["assert_result"] = "found!"
    except AssertionError:
        self.env["assert_result"] = "ERROR: not found!"
```

Recipe Output:

```
com.github.jgstew.SharedProcessors/AssertInputContainsString  
{'Input': {'assert_string': '_', 'input_string': '1_2_3'}}  
AssertInputContainsString: No value supplied for raise_error, sett  
AssertInputContainsString: `1_2_3` must contain `_`  
{'Output': {'assert_result': 'found!'}}
```

Example Recipe: Audacity

```
Description: Downloads the latest version of Audacity for Windows
# based upon https://github.com/autopkg/ahousseini-recipes/blob/main/recipes/Audacity/Audacity-Windows.recipe
Identifier: com.github.jgstew.download.Audacity-Win
Input:
  NAME: Audacity
MinimumVersion: "2.3"
Process:
  - Processor: GitHubReleasesInfoProvider
    Arguments:
      github_repo: audacity/audacity
      asset_regex: .*64bit\.exe$
      include_prereleases: False
  - Processor: URLDownloaderPython
    Arguments:
      filename: "audacity-win-64bit.exe"
```

AutoPkg for Windows Software 1.0

Use AutoPkg on MacOS to handle Windows Software

Started by Matt Hansen at Penn State University in Nov 2014

At the time, AutoPkg could **not** run on Windows

Matt created AutoPkg Processors that would run on MacOS but extract metadata for Windows Software

Presented at PSU MacAdmins in 2015 :

<https://www.youtube.com/watch?v=w4WM6M89hmg>

First:

Added CrashPlan download recipes for additional platforms

 master



hansen-m committed on Nov 12, 2014



Showing 4 changed files with 208 additions and 0 deletions.



Filter changed files



Code42



CrashPlanPROeClientLinu...



CrashPlanPROeClientSolar...



CrashPlanPROeClientWind...



52



C

...

...

1

+ +

2

+ +

3

+ +

4

+ +

Can we?

Create file_meta_data.py

main

v0.2.18 ... 0.1.0



jgstew committed on Mar 5, 2015

Showing 1 changed file with 47 additions and 0 deletions.

47 file_meta_data.py

...

...

@@ -0,0 +1,47 @@

1

+

2

+ import hachoir_core

3

+ import hachoir_core.cmd_line

4

+ import hachoir_metadata

5

+ import hachoir_parser

First Windows Processors:

Added new SharedProcessors

 master



hansen-m committed on Jul 6, 2015



Showing 4 changed files with 269 a



SharedProcessors



HachoirMetaDataPr...



MSIVersionProvider....



WinInstallerExtracto...



hansen-m/HachoirMetadataProvider3

```
com.github.hansen-m.SharedProcessors/HachoirMetadataProvider3
{'Input': {'metadata_key': 'mime_type'}}
HachoirMetadataProvider3: Examining: /Users/jgstew/Library/AutoPkg/Cache/com
vider3/downloads/unzip-5.52.exe
HachoirMetadataProvider3: Metadata:
HachoirMetadataProvider3: - Creation date: 2005-02-28 21:51:12
HachoirMetadataProvider3: - Comment: CPU: Intel 80386
HachoirMetadataProvider3: - Comment: Subsystem: Windows CUI
HachoirMetadataProvider3: - Format version: Portable Executable: Windows app
HachoirMetadataProvider3: - MIME type: application/x-dosexec
HachoirMetadataProvider3: - Endianness: Little endian
HachoirMetadataProvider3: Found Metadata: mime_type = application/x-dosexec
```


these changes will allow autopkg to display its help on
linux/windows...

[Browse files](#)

... autopkg will not function without overloading or replacing some of OS X
specific code, but this is a step in that direction. These changes should
have no effect of autopkg running on OS X.



master (#169)



v2.7.2 ... 2.0b2



jgstew committed on Apr 1, 2015

1 parent [909cc25](#) commit [495e035](#)

Showing 7 changed files with 49 additions and 17 deletions.

Split

Unified

Processor: URLDownloader

Uses cURL to download files

Checks if they have changed before downloading

Stores HTTP Header info using XAttr / Extended File Attributes

XAttr only works on MacOS and Some Linux Filesystems!

Anywhere XAttr does not work, the download will happen every run!!!

AutoPkg for Windows Software 2.0

AutoPkg now runs on Windows, MacOS, and Linux!

URLDownloaderPython Processor is the key! (April 2021)

Project started in full in January 2022

~50 other Processors created as well, all Open Source

Develop Recipes on Windows (or MacOS), automated builds on Ubuntu

Scripts to automate the setup of AutoPkg dev/build environment

Recipes can now be written in YAML!

URLDownloaderPython



Downloads using Python instead of cURL

Can compute hashes while downloading (md5, sha1, sha256)

Stores HTTP headers and other info in a JSON file instead of XAttr

- Works on MacOS, Windows, and Linux regardless of filesystem

URLDownloaderPython: Example JSON Metadata

| Name | Size |
|--|-----------|
|  audacity-win-64bit.exe | 14.3 MB |
|  audacity-win-64bit.exe.info.json | 358 bytes |

URLDownloaderPython: Example JSON Metadata

Users > jgstew > Library > AutoPkg > Cache > com.github.jgstew.download.Audacity-Win > downloads

```
1  {
2      "file_name": "audacity-win-64bit.exe",
3      "file_size": 14281408,
4      "download_url": "https://github.com/audacity/audacity/releases/download/Auda
5      "http_headers": {
6          "Content-Length": 14281408,
7          "ETag": "\"0x8DAA70183E76C21\"",
8          "Last-Modified": "Wed, 05 Oct 2022 18:43:37 GMT"
9      }
10 }
```

URLDownloaderPython: Example with Hashing!

`{}` audacity-win-64bit.exe.info.json ✕

Users > jgstew > Library > AutoPkg > Cache > com.github.jgstew.download.Audacity

```
1  {
2      "file_name": "audacity-win-64bit.exe",
3      "file_size": 14281408,
4      "file_sha1": "3502f604df6c3c006d84c704b16d2ef3b135a26e",
5      "file_sha256": "35a318c716f2579bb81229c04375bc21cac878ad011",
6      "file_md5": "d90609b35ce74471bae445d71c00cecc",
7      "download_url": "https://github.com/audacity/audacity/releases",
8      "http_headers": {
9          "Content-Length": 14281408,
10         "ETag": "\"0x8DAA70183F76C21\""
11     }
```

My Processors:

- 845, jgstew/URLDownloaderPython
- 15, jgstew/BigFixPrefetchItem
- 381, jgstew/DateTimeFromString
- 400, jgstew/FileExeGetInfoPE
- 143, jgstew/FileExeVerifySignature
- 3, jgstew/FileExeVersionExtractor
- 16, jgstew/FileGetExecutable
- 146, jgstew/FileMsiGetInfoOLE
- 142, jgstew/FileMsiGetProperty
- 9, jgstew/FileTextSearcher
- 27, jgstew/StopProcessingIfDownloadUnchanged
- 8, jgstew/TextSearcher
- 2, jgstew/TextSubstitutionRegex
- 35, jgstew/URLTextSearcherArray
- 5, jgstew/VersionGetMajorMinor
- 36, jgstew/VersionMaximumArray

Other Processors:

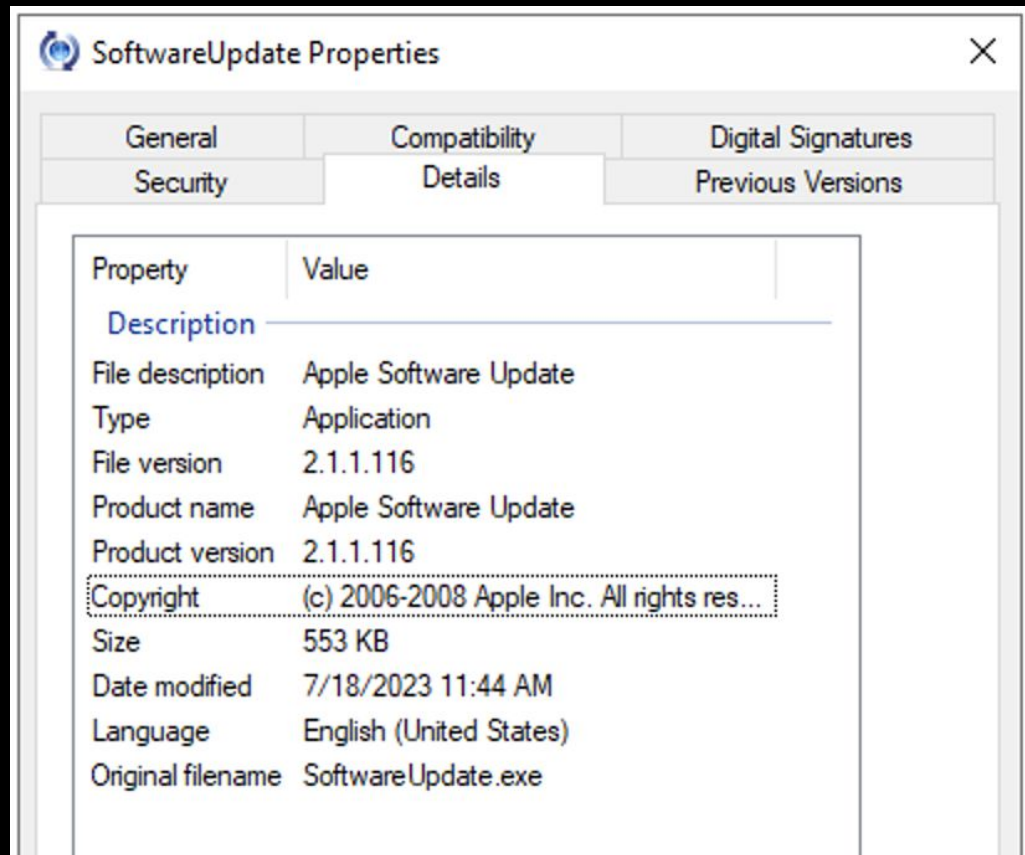
- 300, GitHubReleasesInfoProvider
- 309, URLTextSearcher
- 16, hansen-m/WinInstallerExtractor
- 39, homebysix/FindAndReplace

Processors to extract Metadata from Windows Files

- FileExeGetInfoPE
 - File Version
- FileExeVerifySignature
 - File Signature Date (and validity)
- FileExeVersionExtractor
 - File Version
- FileMsiGetInfoOLE
 - MSI OLE db updated at
- FileMsiGetProperty
 - MSI Software Version

FileExeGetInfoPE

Use python to get this info:



FileExeGetInfoPE

```
com.github.jgstew.SharedProcessors/FileExeGetInfoPE
```

```
{'Input': {}}
```

```
FileExeGetInfoPE: No value supplied for custom_peinfo_index, setting default value of: FileVersion
```

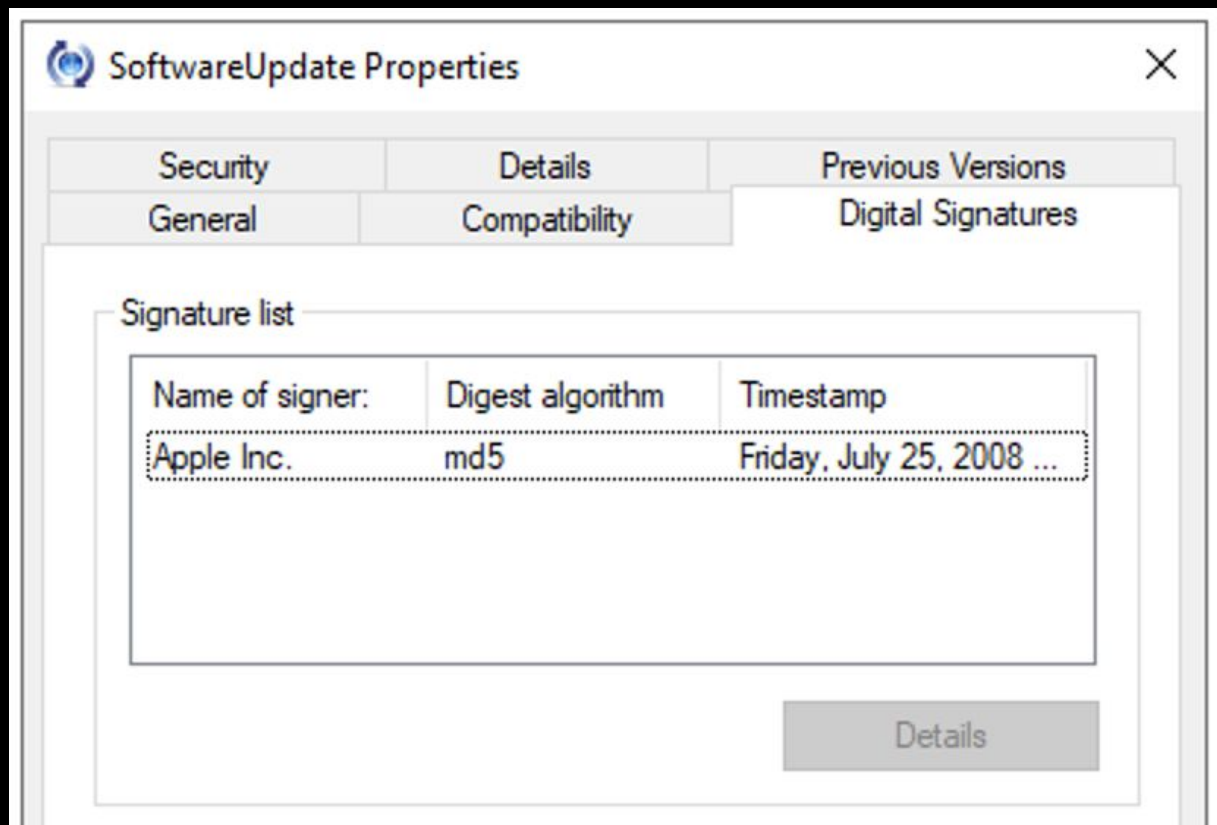
```
FileExeGetInfoPE: No value supplied for custom_peinfo_output, setting default value of: version
```

```
FileExeGetInfoPE: No value supplied for peinfo_first_only, setting default value of: True
```

```
{'Output': {'file_peinfo_CompanyName': 'Apple Inc.',  
            'file_peinfo_FileDescription': 'Apple Software Update',  
            'file_peinfo_FileVersion': '2.1.1.116',  
            'file_peinfo_InternalName': 'Apple Software Update',  
            'file_peinfo_LegalCopyright': '(c) 2006-2008 Apple Inc. All rights reserved.',  
            'file_peinfo_OriginalFilename': 'SoftwareUpdate.exe',  
            'file_peinfo_ProductName': 'Apple Software Update',  
            'file_peinfo_ProductVersion': '2.1.1.116',  
            'version': '2.1.1.116'}}
```

FileExeVerifySignature

Use python to get this:



FileExeVerifySignature

```
com.github.jgstew.SharedProcessors/FileExeVerifySignature
{'Input': {'file_pathname': '/Users/jgstew/Library/AutoPkg/Cache/com.github.jgstew.teds/SoftwareUpdate.exe',
           'file_signature_expected_serial_number': '12451790217711796967571790799059',
           'file_signature_throw_error': True,
           'file_sig_date_custom_output': True},
 'Output': {'SourceReleaseDate': '2008-07-25',
            'file_signature_date': '2008-07-25',
            'file_signature_datetime': '2008-07-25 22:21:53+00:00',
            'file_signature_more_info': 'http://www.apple.com/macosx',
            'file_signature_program_name': 'Apple Software Update',
            'file_signature_result': 'VALID',
            'file_signature_result_raw': 'AuthenticodeVerificationResult.OK',
            'file_signature_serial_number': '12451790217711796967571790799059482938',
            'file_signature_valid': True}}
```

FileMsiGetProperty

```
FileMsiGetProperty: Info: `msilib` not found, assuming non-Windows. Attempting to use `msiinfo` instead.  
FileMsiGetProperty: ERROR: file missing! `/usr/bin/msiinfo`  
FileMsiGetProperty: Info: using msiinfo found here: `/usr/local/bin/msiinfo`  
FileMsiGetProperty: MSI Property `ProductVersion` found: `8.0.12349`  
FileMsiGetProperty: MSI Property `Manufacturer` found: `Microsoft Corporation`  
FileMsiGetProperty: MSI Property `ProductName` found: `Project My Screen App`  
FileMsiGetProperty: MSI Property `ProductCode` found: `{64537E9A-4DAE-42F9-BCD8-8AEEB84D1786}`  
FileMsiGetProperty: MSI Property `UpgradeCode` found: `{F7972F08-E790-4E87-B902-B22458F222B6}`  
FileMsiGetProperty: MSI Property `ProductVersion` found: `8.0.12349`  
FileMsiGetProperty: MSI Property `ProductName` found: `Project My Screen App`
```

Results of our internal AutoPkg project:

- ~ 600 recipes - 90% Windows, 10% Linux, 0% MacOS
- ~ 10,000 content generated with unique software-versions (in 1.5 years)
- ~ 300 recipes in full production use
- ~ 100 recipes disabled due to software or vendor issues
- ~ 30 internal recipe developers contributing

Project Challenges:

- Standardize Development Environment
 - Windows
 - Visual Studio Code
- Setup of Recipe Development Prereqs
 - Documentation
 - Setup scripts
- Teaching GIT best practices
 - Branching
 - Pull Requests
- Teaching Regular Expressions
 - GitHubReleasesInfoProvider: asset_regex
 - URLTextSearcher: re_pattern

Automated Setup of Dev Environment: setup scripts

- Install Homebrew (or Chocolatey on windows)
- Install Python 3.10
- Install Visual Studio Code
- Install Recipe Dependencies
 - 7zip
 - MSITools
 - Libmagic
- Clone AutoPkg Repo
- Install Python Package Requirements



setup_mac.sh



setup_ubuntu.sh



setup_win.bat

Visual Studio Code config:

- `.vscode/extensions.json`
 - Automatically suggest installation of required / helpful extensions
 - Python, YAML, etc.
- `.vscode/settings.json`
 - Set some default settings
- `.vscode/launch.json`
 - Run recipes from the debug menu
- `.vscode/tasks.json`
 - Yamllint - validate yaml syntax
 - Pre-commit - validate all files

Automated Validation of Files: Pre-Commit

Demo

Test-Recipes

Demo

Running in Docker

- XAttr / URLDownloader does not work correctly
 - But URLDownloaderPython does work correctly!
- Example Docker File Provided: (Ubuntu)
 - <https://github.com/jgstew/jgstew-recipes/blob/main/Dockerfile>

Automated Testing of Recipes: GitHub Actions

Demo

Questions?

- Links
 - <https://github.com/jgstew/MacAdmins2023>
- Feedback
 - <https://bit.ly/psumac2023-127>

