How to Upload Packages to Jamf Cloud using Autopkgr

This guide was created to assist others who are looking to upload packages to their Jamf Cloud distribution point using Autopkgr without the need to create smart groups or policies. There are GitHub repositories that contain jss recipes but those recipes will create smart groups and policies. The recipes that we will make in this guide will be jss-upload recipes and will not create smart groups or policies.

Autopkgr will name the package with the version number of the software that you're packaging which will allow you to easily match up the software version number with a patch management definition in Jamf Pro. Autopkgr can be configured to automatically update repositories and run the jss-upload recipes on a schedule making the process of getting updated packages into your Jamf Cloud repository much easier.

This guide will teach you how to create recipes that will use parent recipes from the autopkg GitHub repo. It's intended to be a step by step guide for people that are new to Autopkgr and recipe creation for jss-upload recipes.

Hat tip to Daz Wallace and Graham Pugh for guiding me through the process of understanding how to make autopkg recipes with the minimum requirements for packaging files and uploading them to a Jamf Cloud distribution point.

Daz's guide can be found here:
https://dazwallace.wordpress.com/2019/03/12/using-autopkg-for-package-uploads-to-jamf-cloud-only/

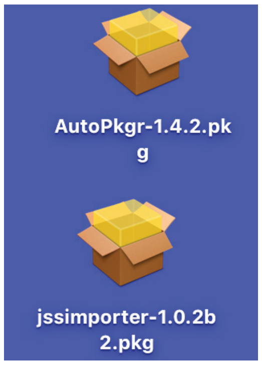
Requirements

1. Install AutoPkgr. Get it here: https://github.com/lindegroup/autopkgr/releases/tag/v1.4.2
2. Install JSSImporter. Get it here: https://github.com/jssimporter/JSSImporter/releases
3. A Jamf Pro cloud hosted server. More info here: https://www.jamf.com/products/jamf-cloud/
4. A plain text editor. I recommend BBEdit. Get a free 30 day trial here: https://www.barebones.com/products/bbedit/download.html
5. A GitHub repository. Create one here: https://github.com

Section 1. Install and configure AutoPkgr and JSSImporter.

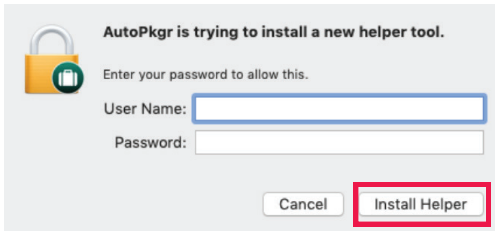
In this section we will discuss the installation and configuration of Autopkgr, JSSImporter and subscribing to the autopkg GitHub repository.

1. Install the AutoPkgr and JSSImporter importer packages, accepting all default dialog windows.



2. Open AutoPkgr located in the Applications folder.

3. Enter your administrative credentials at the message below, then click the Install Helper button.



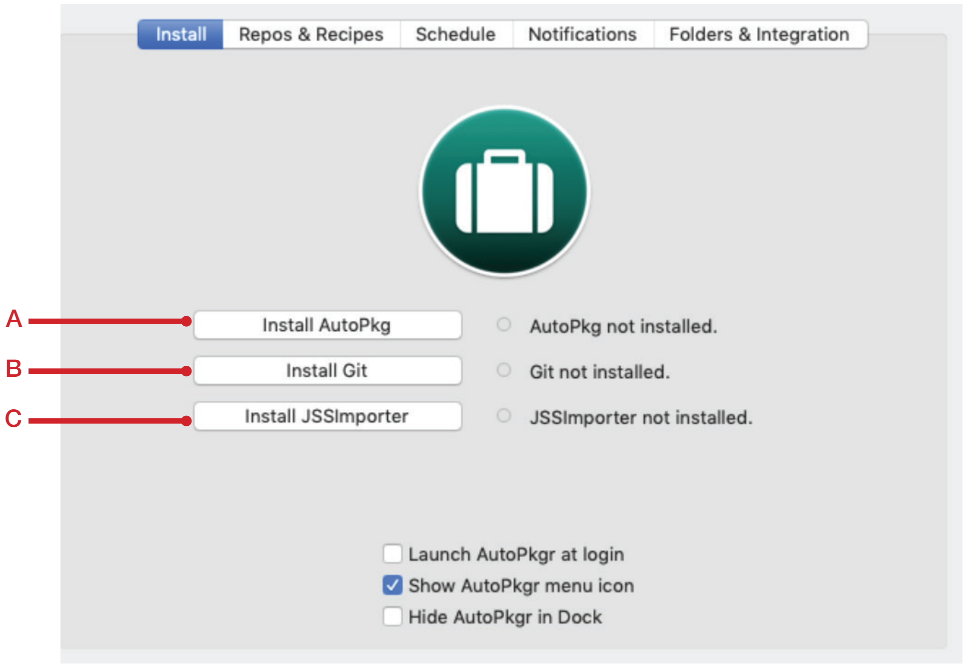
AutoPkgr is trying to install a new helper tool.

Enter your password to allow this.

User Name:

Password:

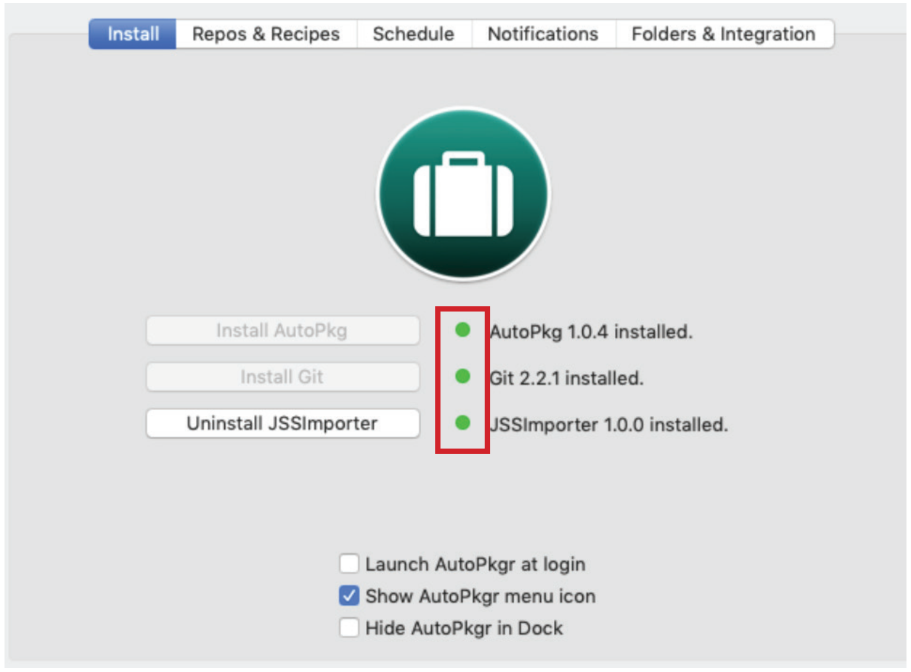
4. Follow these steps:
   A. Click the Install AutoPkg button. Enter your admin credentials when prompted.
   B. Click the Install Git button. Enter your admin credentials when prompted.
   C. Click the Install JSSImporter button. Enter your admin credentials when prompted.



AutoPkg not installed.
Git not installed.
JSSImporter not installed.

☐ Launch AutoPkgr at login
☑ Show AutoPkgr menu icon
☐ Hide AutoPkgr in Dock

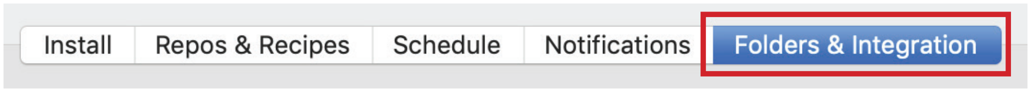
5. Once installed, all items will have green status lights next to them.



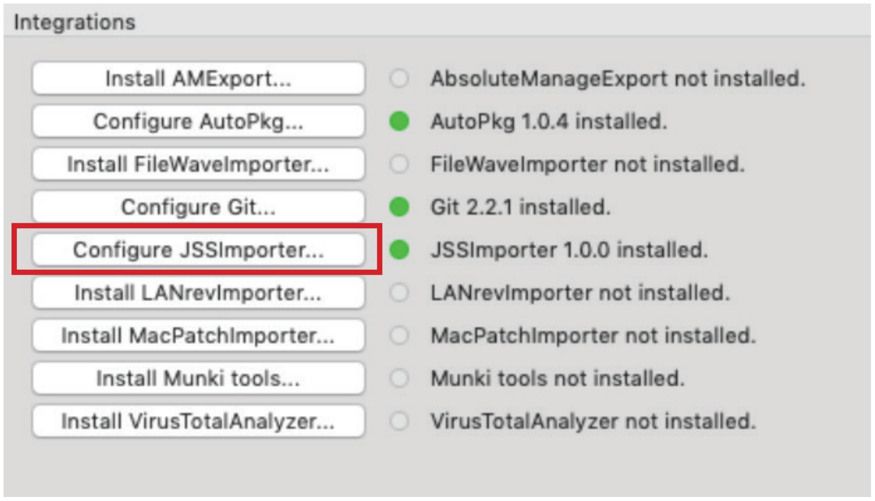
● AutoPkg 1.0.4 installed.
● Git 2.2.1 installed.
● JSSImporter 1.0.0 installed.

☐ Launch AutoPkgr at login
☑ Show AutoPkgr menu icon
☐ Hide AutoPkgr in Dock

6. Select the Folders & Integration tab.

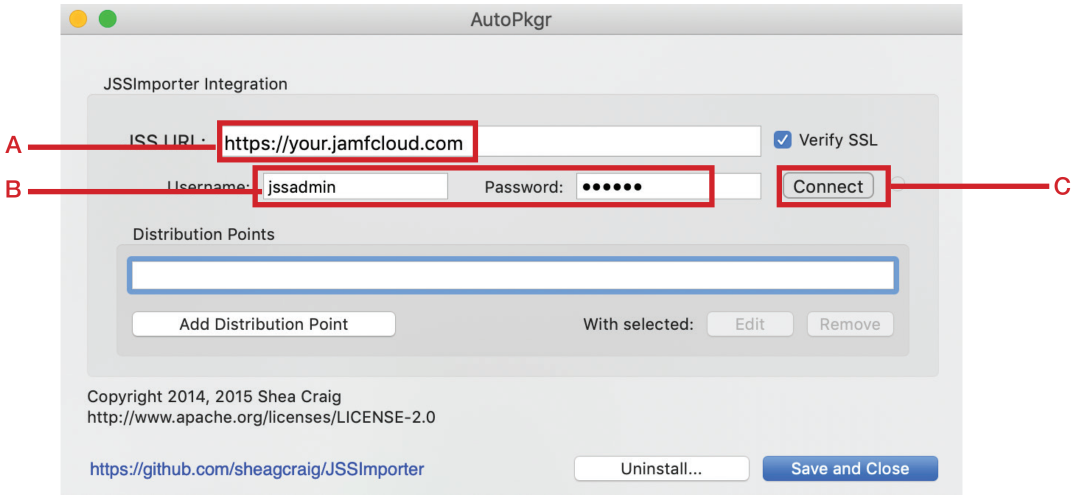


7. Click the Configure JSSImporter button.



Install AMExport...		AbsoluteManageExport not installed.
Configure AutoPkg...	●	AutoPkg 1.0.4 installed.
Install FileWaveImporter...		FileWaveImporter not installed.
Configure Git...	●	Git 2.2.1 installed.
Configure JSSImporter...	●	JSSImporter 1.0.0 installed.
Install LANrevImporter...		LANrevImporter not installed.
Install MacPatchImporter...		MacPatchImporter not installed.
Install Munki tools...		Munki tools not installed.
Install VirusTotalAnalyzer...		VirusTotalAnalyzer not installed.

8. Follow these steps:
   A. Enter the URL of your Jamf Cloud server.
   B. Enter your Jamf Administrative credentials or an account that has proper privileges.
   C. Click the Connect button.

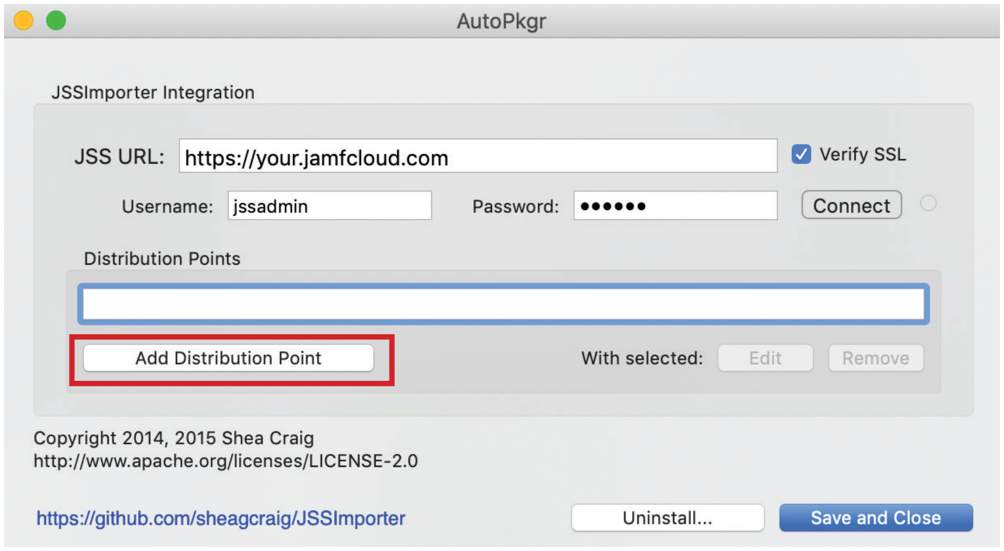


JSS URL: https://your.jamfcloud.com
☑ Verify SSL
Username: jssadmin Password: ••••••

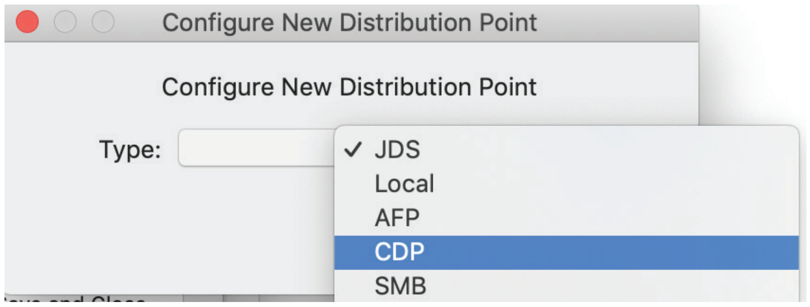
Copyright 2014, 2015 Shea Craig
http://www.apache.org/licenses/LICENSE-2.0

https://github.com/sheagcraig/JSSImporter

9. Click the Add Distribution Point button.

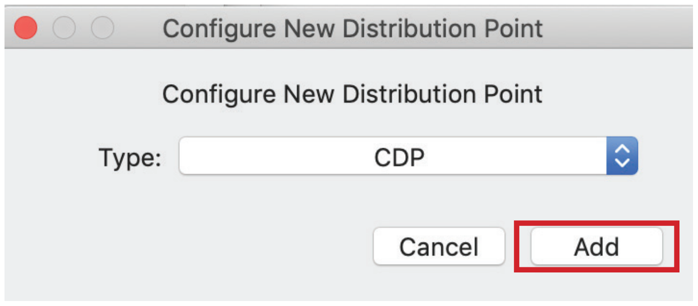


10. Select CDP from the dropdown menu.



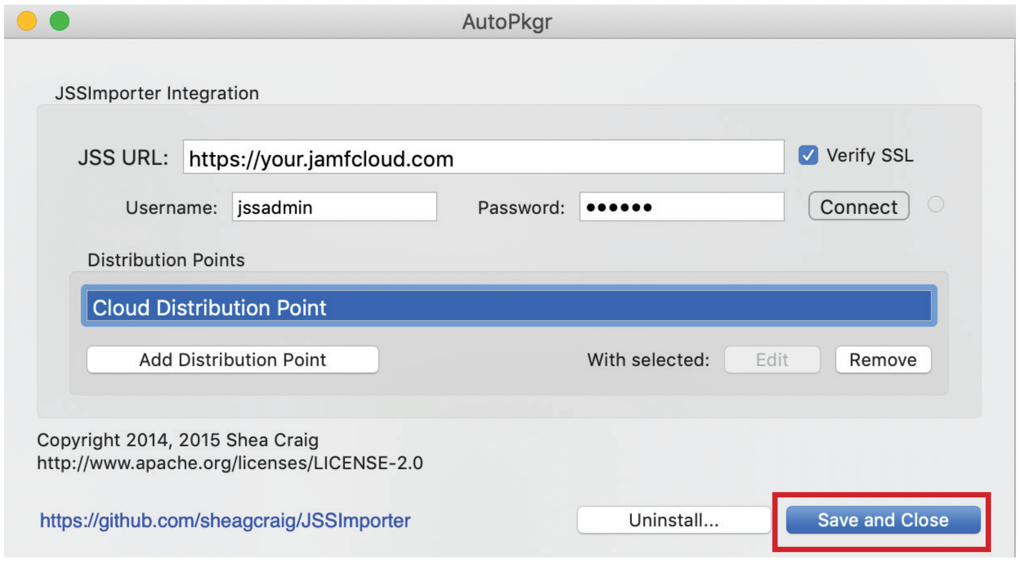
Configure New Distribution Point
Configure New Distribution Point
Type:
✓ JDS
Local
AFP
CDP
SMB

11. Click the Add button.



Configure New Distribution Point
Configure New Distribution Point
Type: CDP
Cancel
Add

12. Click the Save and Close button.



AutoPkgr

JSSImporter Integration

JSS URL: https://your.jamfcloud.com ✓ Verify SSL

Username: jssadmin Password: ●●●●●● Connect

Distribution Points

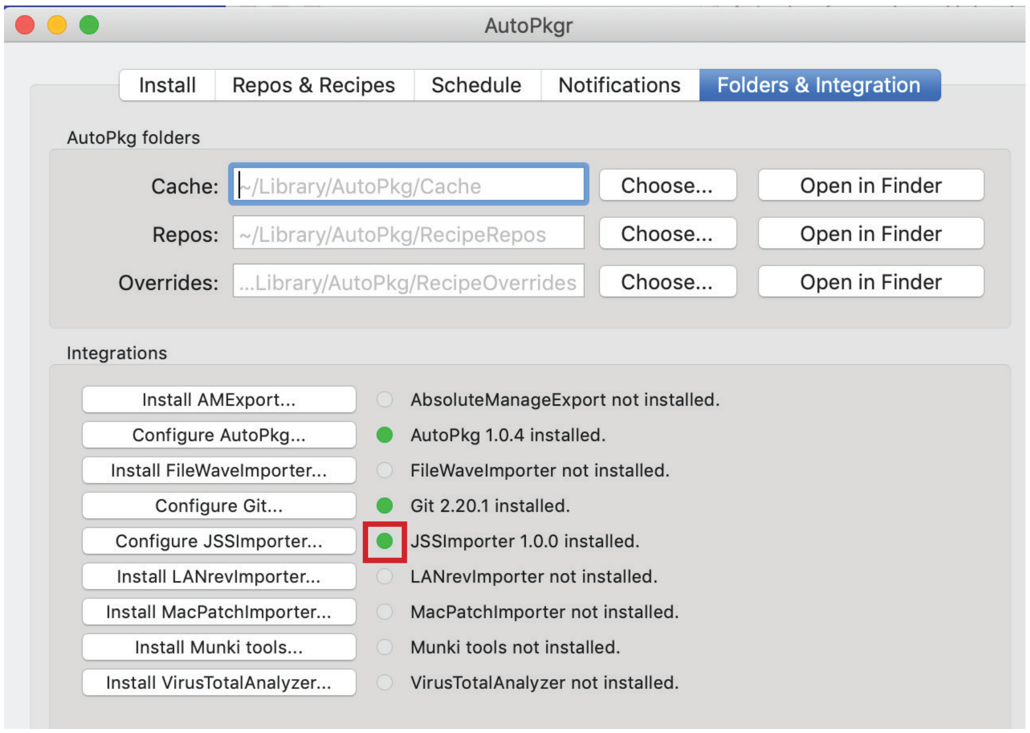
Cloud Distribution Point

Add Distribution Point With selected: Edit Remove

Copyright 2014, 2015 Shea Craig
http://www.apache.org/licenses/LICENSE-2.0

https://github.com/sheagcraig/JSSImporter

13. If everything went well, you will see a green status light next to the Configure JSSImporter button.



AutoPkgr

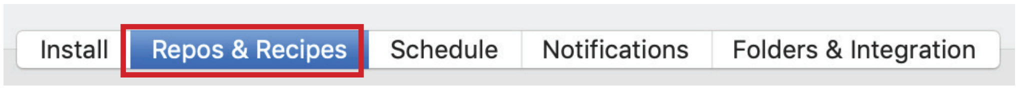
AutoPkg folders

Cache: ~/Library/AutoPkg/Cache
Repos: ~/Library/AutoPkg/RecipeRepos
Overrides: ...Library/AutoPkg/RecipeOverrides

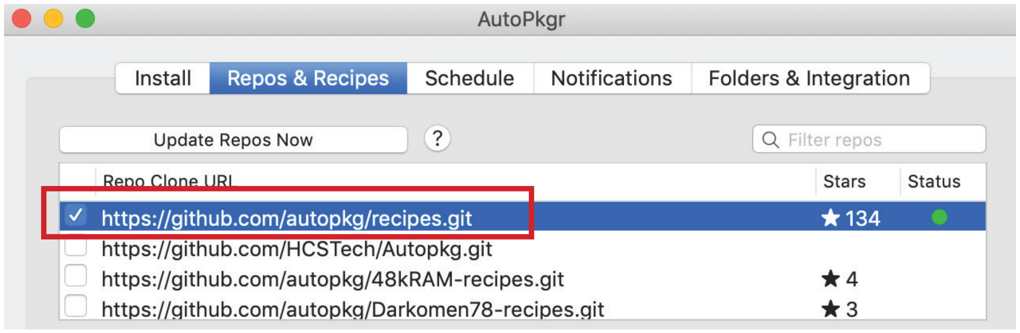
Integrations

Install AMExport...		AbsoluteManageExport not installed.
Configure AutoPkg...	●	AutoPkg 1.0.4 installed.
Install FileWaveImporter...		FileWaveImporter not installed.
Configure Git...	●	Git 2.20.1 installed.
Configure JSSImporter...	●	JSSImporter 1.0.0 installed.
Install LANrevImporter...		LANrevImporter not installed.
Install MacPatchImporter...		MacPatchImporter not installed.
Install Munki tools...		Munki tools not installed.
Install VirusTotalAnalyzer...		VirusTotalAnalyzer not installed.

14. Select the Repos & Recipes tab.

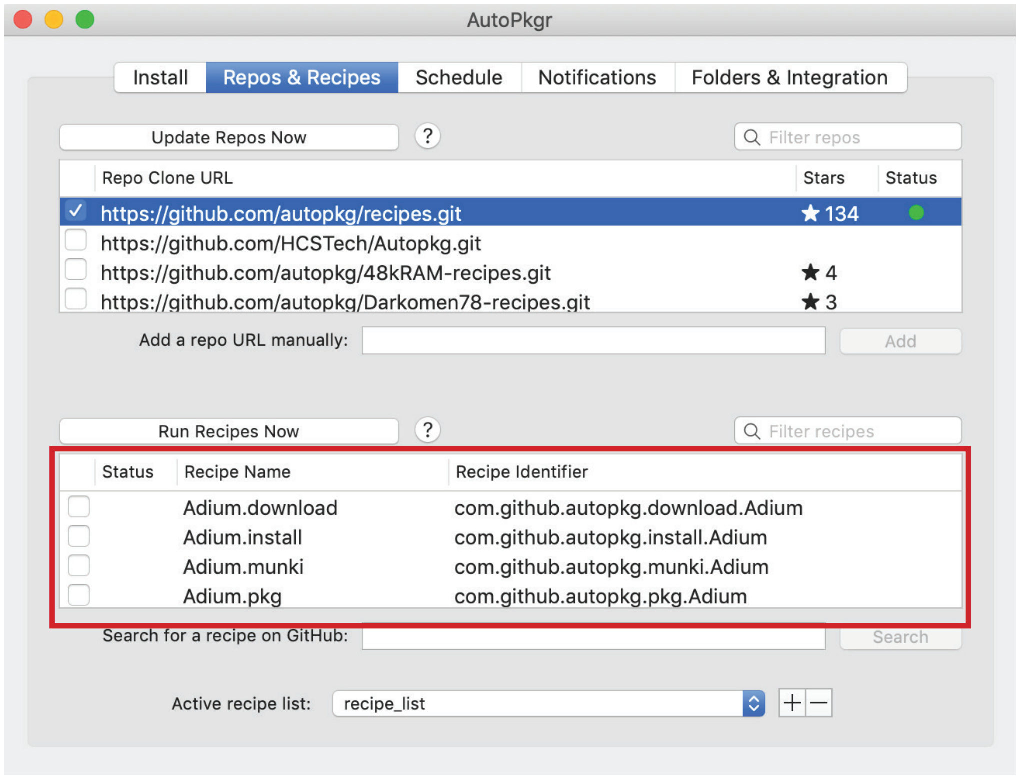


15. In the Repo section, select the autopkg repo. It's named: https://github.com/autopkg/recipes.git



	Repo Clone URL	Stars	Status
✓	https://github.com/autopkg/recipes.git	★ 134	●
	https://github.com/HCSTech/Autopkg.git		
	https://github.com/autopkg/48kRAM-recipes.git	★ 4	
	https://github.com/autopkg/Darkomen78-recipes.git	★ 3	

16. The autopkg recipes will now show up in the recipes section.



	Repo Clone URL	Stars	Status
✓	https://github.com/autopkg/recipes.git	★ 134	●
	https://github.com/HCSTech/Autopkg.git		
	https://github.com/autopkg/48kRAM-recipes.git	★ 4	
	https://github.com/autopkg/Darkomen78-recipes.git	★ 3	

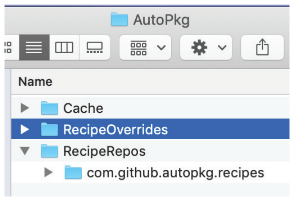
	Status	Recipe Name	Recipe Identifier
		Adium.download	com.github.autopkg.download.Adium
		Adium.install	com.github.autopkg.install.Adium
		Adium.munki	com.github.autopkg.munki.Adium
		Adium.pkg	com.github.autopkg.pkg.Adium

Section 2. Creating a jss-upload recipe

In this section we will discuss the creation of a jss-upload recipe using a parent recipe from the autopkg GitHub repository. Parent recipes do most of the work as they contain all the steps to download and package the software for our jss-upload recipe. We will also cover configuring a manual GitHub repository in Autopkgr.

1. In order to use a parent recipe for our jss-upload recipe, we must be subscribed to the autopkg Github repository in Autopkgr which we did in step 16 in the first section of this document. Once we subscribe to a repository, it will clone it to the AutoPkg folder which is located in the User Library folder. ~/Library/AutoPkg. If that folder was not present our your Mac, you would not be able to reference it as the parent for your jss-upload recipe.

   NOTE: You can use other GitHub repositories if you need to for the parent recipe. I chose to use the autopkg repository as most other repositories use it as their parent.



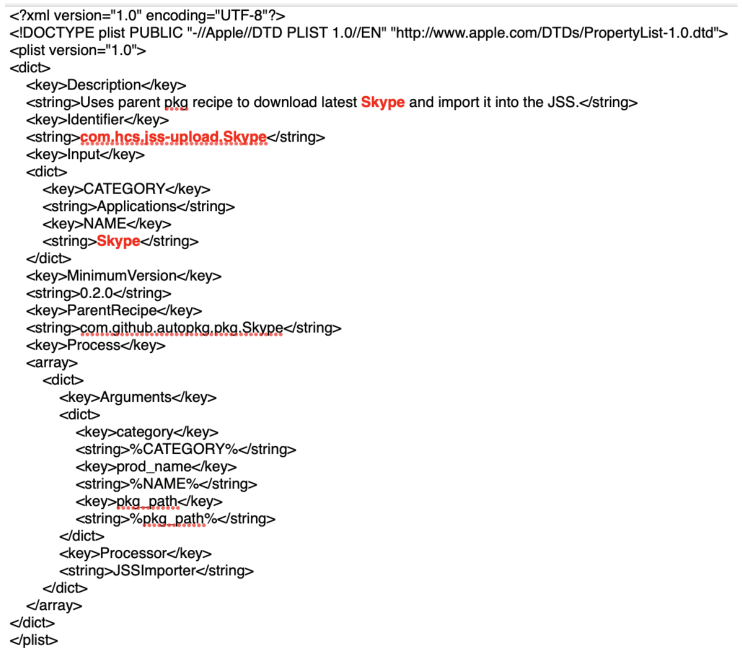
2. This is a sample of a jss-upload recipe file. To follow along with this guide, download this sample recipe file from the HCS GitHub repository here:

   https://github.com/HCSTech/Autopkg/blob/master/Skype-jss-upload.recipe

   Once downloaded, open it with BBEdit or your favorite text editor.

   The items in RED are the items that you would change if you wanted to use this as a template for a different software title. For example, say you want to use this template to pkg and upload Google Chrome to your Jamf Cloud distribution point. For each item named Skype in RED, with an exception of the ParentRecipe, change it to GoogleChrome.

   Now we need to find the Identifier for the ParentRecipe which lives in the autopkg repository. We will do that in the next step.



```
<?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>Uses parent pkg recipe to download latest Skype and import it into the JSS.</string>
   <key>Identifier</key>
   <string>com.hcs.jss-upload.Skype</string>
   <key>Input</key>
   <dict>
      <key>CATEGORY</key>
      <string>Applications</string>
      <key>NAME</key>
      <string>Skype</string>
   </dict>
   <key>MinimumVersion</key>
   <string>0.2.0</string>
   <key>ParentRecipe</key>
   <string>com.github.autopkg.pkg.Skype</string>
   <key>Process</key>
   <array>
      <dict>
         <key>Arguments</key>
         <dict>
            <key>category</key>
            <string>%CATEGORY%</string>
            <key>prod_name</key>
            <string>%NAME%</string>
            <key>pkg_path</key>
            <string>%pkg_path%</string>
         </dict>
         <key>Processor</key>
         <string>JSSImporter</string>
      </dict>
   </array>
</dict>
</plist>
```

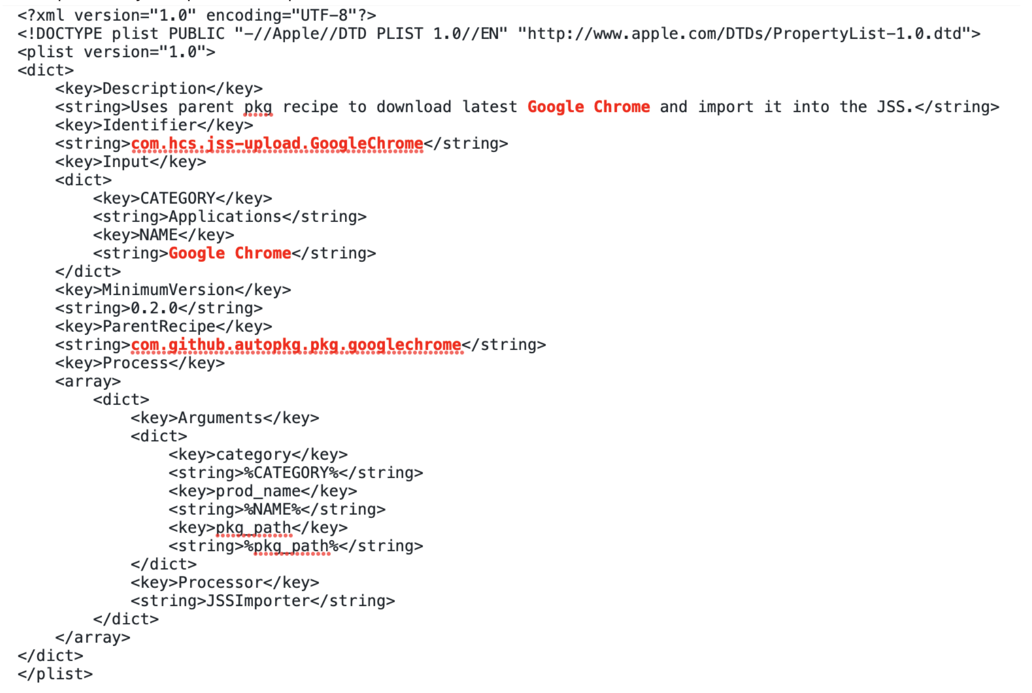
3. Open the Terminal application and enter the following command:
   `autopkg info GoogleChrome.pkg`

   You will see the following output:
   The item in RED is the Identifier. Copy that information then go back to your jss-upload template from step 2 and overwrite the value in the ParentRecipe with this new information. IE.. com.github.autopkg.pkg.googlechrome

```
Description: Downloads latest Google Chrome disk image and builds a package.
Identifier: com.github.autopkg.pkg.googlechrome
Munki import recipe: False
Has check phase: True
Builds package: True
Recipe file path: /Users/work/Library/AutoPkg/RecipeRepos/com.github.autopkg.
recipes/GoogleChrome/GoogleChrome.pkg.recipe
Parent recipe(s): /Users/work/Library/AutoPkg/RecipeRepos/com.github.autopkg.
recipes/GoogleChrome/GoogleChrome.download.recipe
```

   The item in RED is the Identifier. Copy that information then go back to your jss-upload template from step 2 and overwrite the value in the ParentRecipe with this new information. IE.. com.github.autopkg.pkg.googlechrome

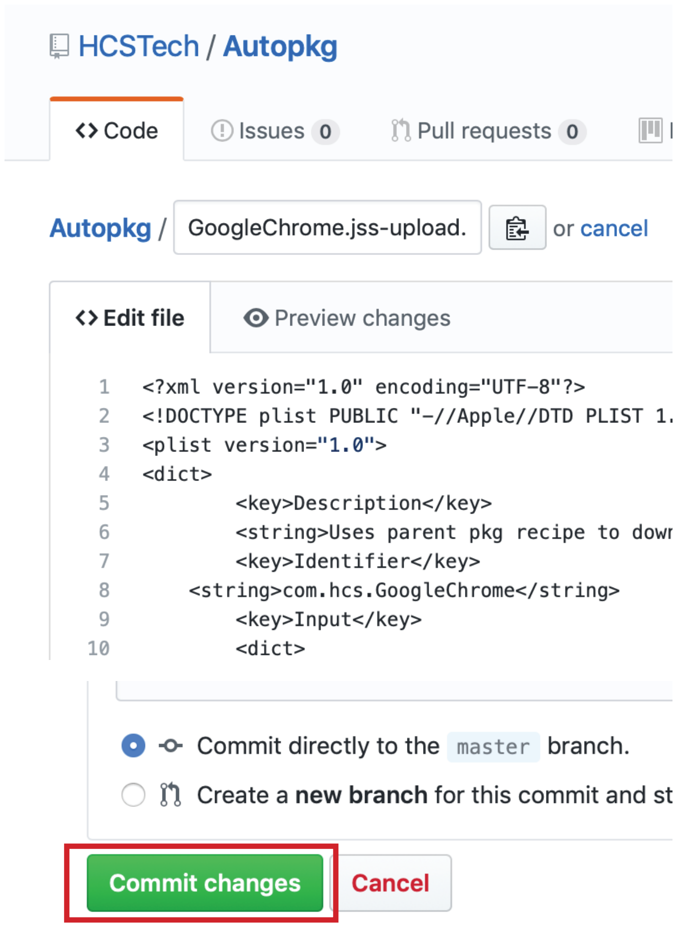
4. The updated jss-upload recipe should look like the item below.



```
<?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>Uses parent pkg recipe to download latest Google Chrome and import it into the JSS.</string>
    <key>Identifier</key>
    <string>com.hcs.jss-upload.GoogleChrome</string>
    <key>Input</key>
    <dict>
        <key>CATEGORY</key>
        <string>Applications</string>
        <key>NAME</key>
        <string>Google Chrome</string>
    </dict>
    <key>MinimumVersion</key>
    <string>0.2.0</string>
    <key>ParentRecipe</key>
    <string>com.github.autopkg.pkg.googlechrome</string>
    <key>Process</key>
    <array>
        <dict>
            <key>Arguments</key>
            <dict>
                <key>category</key>
                <string>%CATEGORY%</string>
                <key>prod_name</key>
                <string>%NAME%</string>
                <key>pkg_path</key>
                <string>%pkg_path%</string>
            </dict>
            <key>Processor</key>
            <string>JSSImporter</string>
        </dict>
    </array>
</dict>
</plist>
```

5.This step requires a GitHub repository. Log in to your GitHub repository. Create a new file and name it: GoogleChrome.jss-upload.recipe. Paste in a copy of the Google Chrome recipe that you created in step 4 then commit the changes.

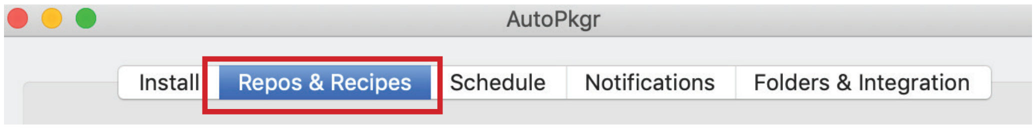
NOTE: Screen shot truncated to fit this guide.



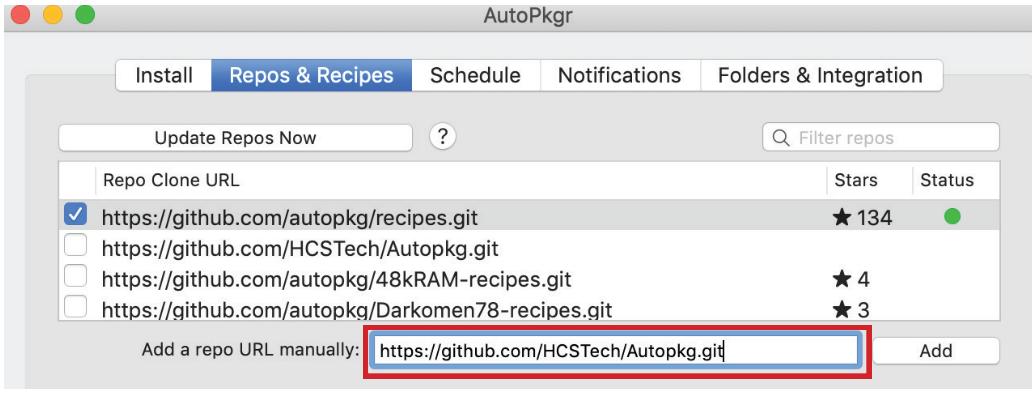
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.
<plist version="1.0">
<dict>
        <key>Description</key>
        <string>Uses parent pkg recipe to dowr
        <key>Identifier</key>
    <string>com.hcs.GoogleChrome</string>
        <key>Input</key>
        <dict>
```

[✓ Commit directly to the master branch.]
Create a **new branch** for this commit and st

6. Open AutoPkgr, then select the Repos & Recipes tab.



7. Go to the Add a repo URL manually: field, and enter the URL of your GitHub repo then Click the Add button.

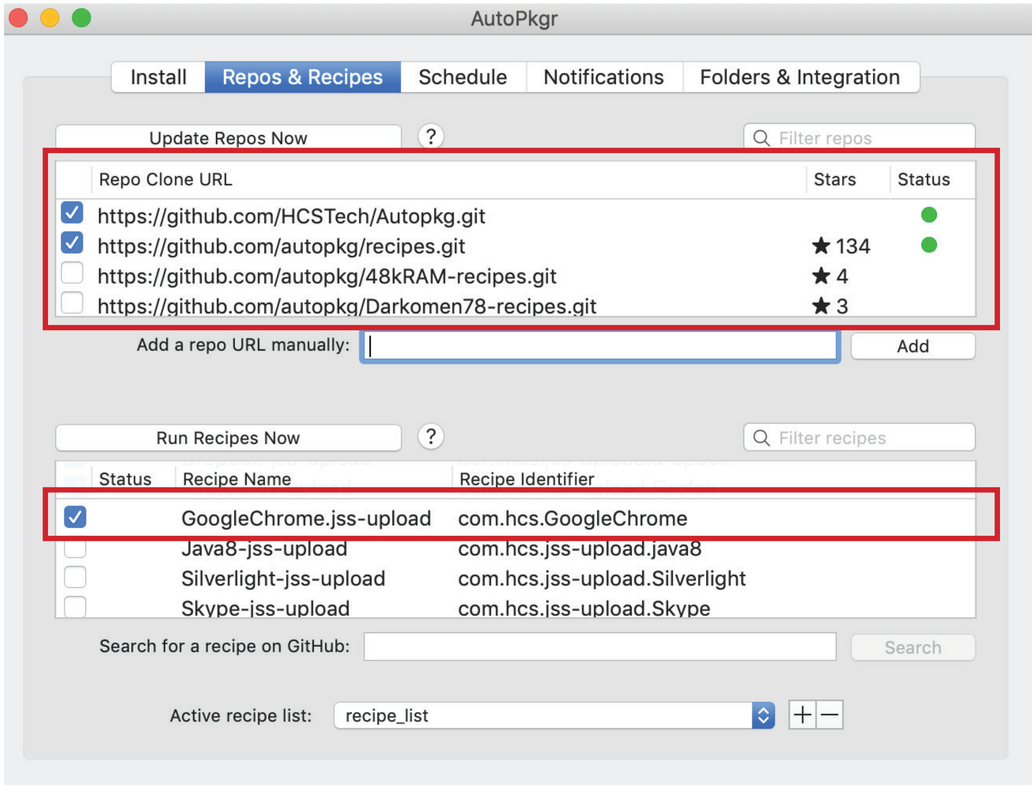


	Repo Clone URL	Stars	Status
✓	https://github.com/autopkg/recipes.git	★ 134	●
	https://github.com/HCSTech/Autopkg.git		
	https://github.com/autopkg/48kRAM-recipes.git	★ 4	
	https://github.com/autopkg/Darkomen78-recipes.git	★ 3	

Add a repo URL manually: https://github.com/HCSTech/Autopkg.git

8. Your GitHub repo will be selected and the GoogleChrome.jss-upload recipe will be available in the recipes list. Make sure to put a check in the box next to the GoogleChrome.jss-upload recipe.

NOTE: If you don't see your google chrome recipe, you can search for it in the Filter recipes search box.



	Repo Clone URL	Stars	Status
✓	https://github.com/HCSTech/Autopkg.git		●
✓	https://github.com/autopkg/recipes.git	★ 134	●
	https://github.com/autopkg/48kRAM-recipes.git	★ 4	
	https://github.com/autopkg/Darkomen78-recipes.git	★ 3	

Add a repo URL manually:

	Status	Recipe Name	Recipe Identifier
✓		GoogleChrome.jss-upload	com.hcs.GoogleChrome
		Java8-jss-upload	com.hcs.jss-upload.java8
		Silverlight-jss-upload	com.hcs.jss-upload.Silverlight
		Skype-jss-upload	com.hcs.jss-upload.Skype

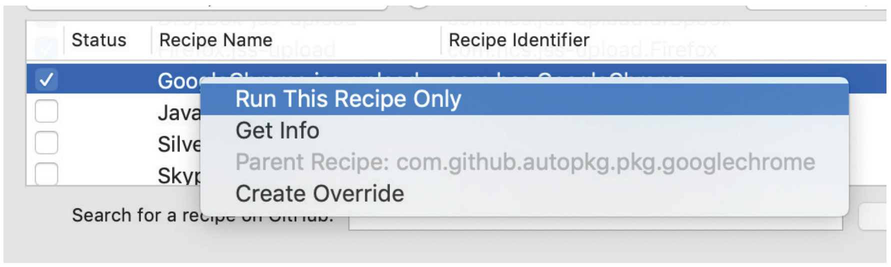
Search for a recipe on GitHub:

Active recipe list: recipe_list

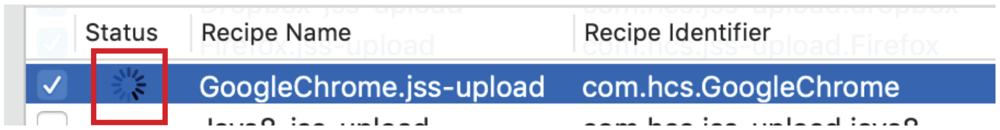
9. Right click on the GoogleChrome.jss-upload recipe and select Run This Recipe Only.

   NOTE: Although they are not required, I would always recommend creating recipe overrides. Without them, you don't know if the parent recipes have been compromised and you could end up uploading something malicious to your repository. It will provide added security. Learn more about recipe overides here:

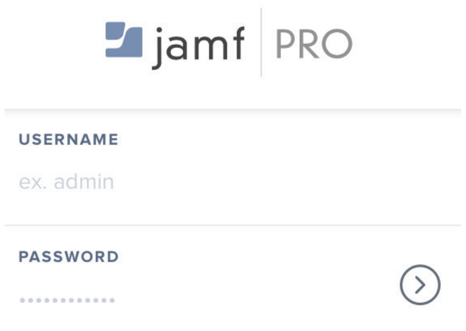
   https://github.com/autopkg/autopkg/wiki/Recipe-Override


Status
Recipe Name
Recipe Identifier
Run This Recipe Only
Get Info
Parent Recipe: com.github.autopkg.pkg.googlechrome
Create Override
Search for a recipe on GitHub:

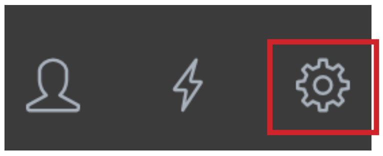
10. You will see a spinning status gear while the recipe is running. When that gear goes away, the upload process has finished.



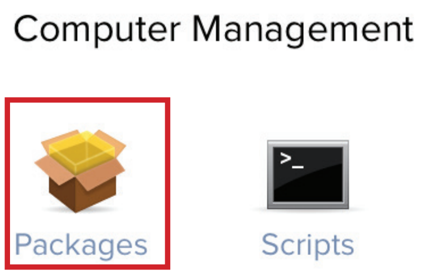
11. Log into your Jamf Pro cloud server.



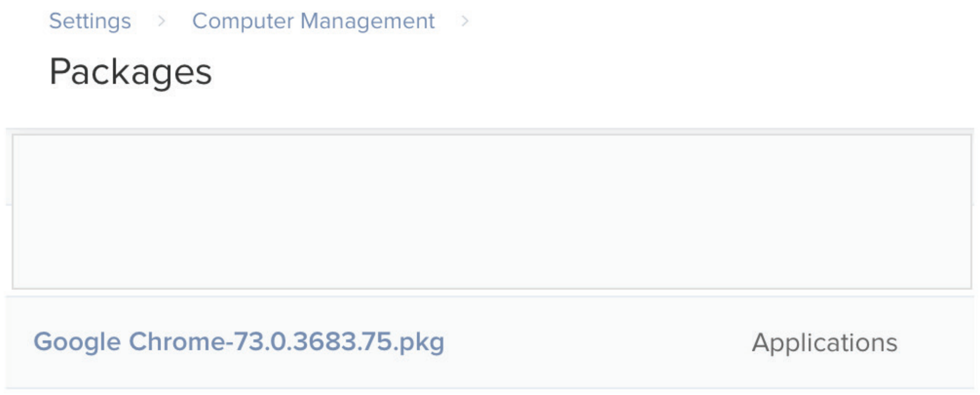
12. Select the settings gear icon in the upper right hand corner.



13. Select the Computer Management section, then select Packages.



14. If all went well, the Google Chrome package will now be in your Jamf Cloud Server repository with a category of Applications.



Google Chrome-73.0.3683.75.pkg	Applications

This completes this guide.