Imaging is Dead
now what?

Greg Neagle, Walt Disney Animation Studios
Imaging is dead.
dying.
no longer recommended.
Imaging will be dead (soon-ish)

I don’t normally try to foretell the future but there is one change for Mac admins that I’m pretty sure will happen:

The coming of Apple File System (APFS) will mark the end of disk imaging on Macs.

For those not familiar with disk imaging, a disk image is a computer file containing the contents and structure of a disk volume. Mac disk images are applied to hard drives using the Apple Software Restore (asr) command line utility to erase the destination drive and then block-copy the data from the disk image onto the destination drive.

Mac deployment practices have generally fallen into one of three categories:

Monolithic imaging

Monolithic imaging is the practice of building a Mac with the desired operating system, desired software, and desired configuration settings, then creating a disk image which includes all the contents of that Mac’s boot drive, including the operating system, installed software, and settings.

Once that disk image is created, the image is then applied to multiple other Macs to make them just like the original Mac.

Modular imaging

Modular imaging is the practice of creating a disk image that contains only the base OS (as well as necessary OS updates from Apple).
RESTORING WITH APFS FILESYSTEMS

Individual APFS volumes can not be restored directly, because their device nodes don't allow I/O from a standard process. However, asr can restore entire APFS containers, including all volumes. Or it can restore valid system configurations, which can get the effect of restoring a single system. This requires understanding what is meant by a valid system.

In order for an APFS volume to be bootable, it must contain a properly installed macOS system. It must also be part of an APFS container which also has two special volumes in it: a Preboot volume and a Recovery volume. A container may have arbitrarily many system volumes in it, but it must have only one Preboot volume and one Recovery volume, each with the corresponding APFS volume role set (see diskutil(1) for information on roles). The Preboot and Recovery volumes contain information which is to each system volume in the container. So for a system volume to be bootable, that information needs to be set up in the Preboot and Recovery volumes. A system which is part of a container that has these two special volumes, and for which the requisite information is set up in those volumes, will be referred to here as a valid system.

If the source of a restore is an APFS image (i.e. an image which contains an APFS container), then asr does different things depending on how target was specified:

Volume Restore
If the target is an individual volume within an existing APFS container, then asr will block restore the APFS container to a file within that volume, after which it will invert the volume within the restored container, erasing the previous contents of the target volume and replacing them with the source volume contents. If the source container only has a single non-special volume (i.e. not Preboot or Recovery), then that is the volume which will be inverted. If the source container has more than one non-special volume, then either the --sourcevolname or --sourcevolumeUUID option must be present and must specify the volume to invert. Additionally, if the volume being inverted is a valid system (as defined above), then the relevant contents of both the Preboot and Recovery volumes will be copied from the source to the target, creating those volumes on the target if necessary.

Volume Restore with Creation
If the target is a synthesized APFS whole disk or Apple_APFS partition, and the --erase option is not present, then asr will create a new volume in the given container, after which it will do a volume restore to that new volume, as with the previous section. All other volumes in the container are preserved.

Volume Restore with Erase
If the target is a synthesized APFS whole disk or any disk partition, and the --erase option is present, then asr will erase the existing partition, create a new APFS container and a new volume in it, after which it will do a volume restore to that new volume, as with the previous section.
AutoDMG 1.8

MagerValp released this on Sep 28, 2017 - 6 commits to master since this release

Assets

- AutoDMG-1.8.dmg
- Source code (zip)
- Source code (tar.gz)

- Added 10.13 and APFS support.
- Added APFS/HFS+ selection to the Advanced Options window.
- Added -F /--filesystem option to CLI.
- Added Filesystem option to templates.
- Removed 10.9 support.
- Fixed crash when CLI build can’t eject image.
- Fixed crash when unchecking updates without a source.
- Fixed Unicode handling of paths for additional packages.
This release of Imagr provides support for restoring APFS images. Please review the wiki page for more information on the current caveats for how to build your NBI.

Fixed in this release:

- First boot items weren’t run in order on APFS volumes (#185)
- Imagr would crash when a package failed to install during the Imagr run (#187)

Imagr 1.3.9: Imaging isn't dead!

grahamgilbert released this on Sep 13, 2017 - 70 commits to master since this release
Category: Downloads

DeployStudio Server 1.7.8

Posted on 06/09/2017 by sadmin

Bug fixes Runtime – GUI deadlock after authentication. Assistant – Sierra and older based NetBoot sets creation. Notes – ASR multicast is not reliable in macOS 10.8 so keep your server running macOS 10.7.5 or upgrade to macOS 10.12. – …

Read more ›

DeployStudio Server 1.7.7

Posted on 30/08/2017 by sadmin

What's new Runtime – preliminary support for macOS 10.13 and APFS imaging. As monolithic imaging is no more supported by Apple, use it carefully (https://support.apple.com/en-us/HT208020). Assistant – new server security option to allow third party apps to add computers to …

Read more ›
High Sierra Notes
Erik Gomez edited this page on Oct 26, 2017 - 13 revisions

These notes were based on an imaging workflow that is not officially supported by Apple. Please see the startosinstall notes page for an Apple approved method of installing High Sierra.

The current release of Imagr supports running in macOS 10.13 High Sierra and restoring APFS volumes. However, the following should be noted when testing.

Important! When you restore an APFS disk image it will wipe the entire device, not just the partition you specify. For most users, this won't be an issue, but you should be aware of this if you use Imagr with multiple partitions.

What needs testing?

The following still needs extensive testing on macOS High Sierra (10.13):

- Restore both HFS and APFS images
To be able to boot from an APFS formatted disk, the target Mac's firmware needs to be updated. An update package can be generated using the following script (thanks to Greg Neagle and Pepijn Bruinene for their work on this).

The script requires a copy of High Sierra in your /Applications folder. A full .app is required (not a stub installer).

```
#!/bin/sh
# Based on investigations and work by Pepijn Bruinene
# Expects a single /Applications/Install macOS High Sierra.app on disk

IDENTIFIER=com.foo.FirmwareUpdateStandalone
VERSION=1.0

# find the Install macOS High Sierra.app and mount the embedded InstallESD disk in
# echo "Mounting High Sierra ESD disk image..."
/usr/bin/hdiutil mount /Applications/Install\ macOS\ High\ Sierra.app/Contents/Server\ Support/Install\ ESD\ Disk

# expand the FirmwareUpdate.pkg so we can copy resources from it
# echo "Expanding FirmwareUpdate.pkg"
/usr/bin/pkgutil --expand /Volumes/InstallESD/Packages/FirmwareUpdate.pkg /tmp/FirmwareUpdateStandalone

# we don't need the disk image any more
# echo "Ejecting disk image..."
/usr/bin/hdiutil eject /Volumes/InstallESD

# make a place to stage our pkg resources
/bin/mkdir -p /tmp/FirmwareUpdateStandalone/scripts

# copy the needed resources
# echo "Copying package resources..."
/bin/cp /tmp/FirmwareUpdate/Standalone/Resources/complex_scripts/postinstall_actions/update /tmp/FirmwareUpdateStandalone

# add an exit 0 at the end of the script
echo "" >> /tmp/FirmwareUpdateStandalone/scripts/postinstall_actions/update
```
The resulting package (located at
/tmp/FirmwareUpdateStandalone/FirmwareUpdateStandalone.pkg) should be used in a
package component that is installed during the Imagr run (so first_boot should be
false/).

```xml
<dict>
  <key>first_boot</key>
  <false/>
  <key>type</key>
  <string>package</string>
  <key>url</key>
  <string>http://imagr/pkgs/FirmwareUpdateStandalone.pkg</string>
</dict>
<dict>
  <key>type</key>
  <string>image</string>
  <key>url</key>
  <string>http://imagr/dmgs/macOS_10.13.apfs.dmg</string>
</dict>
```

**Disk format**

You should format any existing HFS disks as APFS using Disk Utility before restoring an APFS
disk image into it.
So imaging is not dead!
Thank you!
Questions?
and yet...
Upgrade macOS on a Mac at your institution

If you’re the system administrator for your business or education institution, use these methods to upgrade macOS High Sierra.

Before you upgrade

You must be connected to the Internet when you upgrade your macOS. After your Mac confirms your connection, the Installer uses the model number of your Mac to locate and download a firmware update specific to only that Mac.

Only the macOS Installer can download and install the firmware update. Firmware updates can't be done on external devices, like those connected via Target Disk Mode, Thunderbolt, USB, or Firewire.

Upgrade macOS

You can use these supported methods to install macOS:

- Use the macOS Installer.
- Create a bootable installer. Learn how to Create a bootable installer for macOS.
- Startup from macOS Recovery and install macOS.
- Use System Image Utility and create a NetInstall image.

Installing macOS High Sierra on a Mac that is connected by Target Disk Mode isn't a supported installation method.

Learn more about changes to secure kernel extensions in macOS High Sierra.
About monolithic system imaging

Apple doesn't recommend or support monolithic system imaging when upgrading or updating macOS.

Monolithic system imaging can only be used to re-install macOS, not to upgrade to a new macOS version.

If you try to use a monolithic system image, required firmware updates will be missing from the installation. This causes the Mac to operate in an unsupported and unstable state. You can use system images to re-install the existing operating system on a Mac.

If you need to create an image of an APFS container, use one of these supported methods:
- Disk Utility
- diskutil
- System Image Utility

NetRestore in System Image Utility

If you want to create a NetRestore image that you can use to re-install macOS High Sierra on a Mac with all flash storage, don’t create the image from the macOS Installer. Instead, connect another Mac with macOS High Sierra via Target Disk Mode, and use it as the image source. The resulting NetRestore image will be APFS, and is only supported to re-install macOS High Sierra on a Mac with all flash storage.
About monolithic system imaging

Apple doesn't recommend or support monolithic system imaging when upgrading or updating macOS.

Monolithic system imaging can only be used to re-install macOS, not to upgrade to a new macOS version.

If you try to use a monolithic system image, required firmware updates will be missing from the installation. This causes the Mac to operate in an unsupported and unstable state. You can use system images to re-install the existing operating system on a Mac.

If you need to create an image of an APFS container, use one of these supported methods:

- Disk Utility
- diskutil
- System Image Utility

NetRestore in System Image Utility

If you want to create a NetRestore image that you can use to re-install macOS High Sierra on a Mac with all flash storage, don't create the image from the macOS Installer. Instead, connect another Mac with macOS High Sierra via Target Disk Mode, and use it as the image source. The resulting NetRestore image will be APFS, and is only supported to re-install macOS High Sierra on a Mac with all flash storage.

Published Date: Oct 13, 2017
How to install macOS at your organization

If you're the system administrator for your organization, you can choose from a variety of macOS installation methods.

Before you install, make sure that your Mac is connected to the Internet, which allows installation of both macOS and any firmware updates available for your Mac.

If your Mac is enrolled in the Device Enrollment Program (DEP), you can install macOS using the Software Update commands.

If you're installing from and to your computer's startup disk, you can use the `startosinstall` command, which is in the Contents/Resources directory of the macOS installer. For more information about `startosinstall`, add the `--usage` flag to the command. For example, you can:

- Add the `--installpackage` flag to specify other packages to install after macOS. To build these packages, use the `productbuild` command.
- Add the `--eraseinstall` flag to erase your current startup disk first—if you're installing macOS High Sierra 10.13.4 or later and the volume is part of an APFS container. All volumes in that container are erased during this process.

These other installation methods work whether or not macOS is currently installed on the target volume:

- Install from macOS Recovery.
- Install from the App Store.
- Install using a bootable installer.

Apple doesn't recommend or support monolithic system imaging as an installation method, because the system image might not include model-specific information such as firmware updates.

Published Date: May 10, 2018
How to install macOS at your organization

If you're the system administrator for your organization, you can choose from a variety of macOS installation methods.

Before you install, make sure that your Mac is connected to the Internet, which allows installation of both macOS and any firmware updates available for your Mac.

If your Mac is enrolled in the Device Enrollment Program (DEP), you can install macOS using the Software Update commands.

If you're installing from and to your computer's startup disk, you can use the `startosinstall` command, which is in the Contents/Resources directory of the macOS installer. For more information about `startosinstall`, add the `--usage` flag to the command. For example, you can:

- Add the `--installpackage` flag to specify other packages to install after macOS. To build these packages, use the `productbuild(1)` command.
- Add the `--eraseinstall` flag to erase your current startup disk first—if you're installing macOS High Sierra 10.13.4 or later and the volume is part of an APFS container. All volumes in that container are erased during this process.

These other installation methods work whether or not macOS is currently installed on the target volume:

- Install from macOS Recovery.
- Install from the App Store.
- Install using a bootable installer.

Apple doesn't recommend or support monolithic system imaging as an installation method, because the system image might not include model-specific information such as firmware updates.
Imaging methods that bypass the macOS Installer are limited and not preferred, because important firmware updates won’t be installed. This will cause the Mac to operate in an unsupported and unusable state.

Like iPhone or iPad, Mac computers often rely on firmware updates specific to their model. Similarly, updates to the Mac operating system mandate these firmware updates be installed directly from Apple. The most reliable deployment strategy is to use the macOS Installer. You download the macOS High Sierra Installer and install it in place. You can do this:
A critical software update is required for your Mac. To install this update you need to connect to a network. Select a Wi-Fi network below, or click Other Network Options to connect to the internet using other network devices.
A critical software update is required for your Mac, but an error was encountered while installing this update.

Your Mac can’t be used until this update is installed.

Shut Down  Try Again
The all-new, modular Mac Pro with BrainWaveID
Prepare devices

Restore iMac Pro

In certain circumstances, such as a power failure during a macOS upgrade, an iMac Pro may become unresponsive and must be restored. The requirements for doing an iMac Pro restore are:

- Apple Configurator 2.6 installed on any Mac that supports macOS High Sierra and has Internet access.
  
  You may need to configure your web proxy or firewall ports to allow all network traffic from Apple devices to Apple's network 170.0.0.0/8.

- Supported USB-A or USB-C to USB-C or a Thunderbolt cable, such as the one sold by Apple.
  
  The USB-C cable:
  - Must support both power and data.
  - Can be used in conjunction with a supported Thunderbolt to USB-C adapter, such as the one sold by Apple.

Setup the Mac for restoring the iMac Pro

Prepare the iMac Pro for the restore

Use Apple Configurator 2 to restore the iMac Pro

Was this help page useful? Send feedback.
Do you want to restore and update "iBridge" to the latest version?

You cannot undo this action.

Cancel  Restore
Install instead of Image
Setup workflows
Monolithic imaging
Image Creation Techniques

Macworld SF 2007
Session IT821
About Us

Justin Elliott
Penn State University
Senior Research Programmer
MacEnterprise.org steering committee member

Greg Neagle
Disney Animation
Senior Systems Engineer
MacEnterprise.org steering committee member
Build your perfect box

- Install OS onto model volume
- Create default local accounts
- Install applications
- Configure and customize
Clean up the model

- Delete .lproj, cache files, swap, logs, etc.
- Remove any extra users
- Clean up default users’ home dirs
- Don’t forget to empty the trash!
- Cleanup might be easier booted from the Work volume
Clean up the model

- Consider capturing your final cleanup tasks as a script
- Ensure consistency!

```bash
# clean up global caches and temp data
rm -rf /Library/Caches/*
rm -rf /System/Library/Caches/*
rm -rf /Users/Shared/*
rm -f /private/etc/ssh_host*
```
Create snapshot of model

What tool to use?

Disk Utility

- Builds ASR-restorable images
- Included with Mac OS X
- Recommended by 2 out of 2 presenters

/Applications/Utilities/Disk Utility.app
Create snapshot of model

Make sure permissions are enabled

- Boot from your Work volume
- Select ‘Model’ volume in Finder
- Get Info, “Ignore ownership...” should not be checked.
Prepare the image for deployment

Disk Utility

- Scan the image for restore
- Adds checksums to image to verify restored data
Testing your image

Fixing bigger image issues

- Reboot from your Model volume
- Make your changes
- Clean up your model
- Create snapshot of model
- Test!

(does this sound familiar?)
Monolithic imaging
Automated, modular imaging
Modular imaging

Maintaining machines
Thin imaging

Maintaining machines
No imaging

Maintaining machines
Installing

Maintaining machines
Installation-based deployment workflows
Install instead of Image
macOS High Sierra

To set up the installation of macOS High Sierra, click Continue.
System Image Utility: NetInstall
/System/Library/CoreServices/Applications
System Image Utility

Choose a source

Choose an applicable source that will be used to create a network disk image.

Source: Package-only (no OS) installation

Create a NetInstall which does not install macOS.
Choose a source

Choose an applicable source that will be used to create a network disk image.

Source: Package-only (no OS) installation
- Install macOS High Sierra
- Define NetRestore Source(s)

Customize
Previous
Next
Choose a source

Choose an applicable source that will be used to create a network disk image.

Source: Install macOS High Sierra
macOS 10.13.5 (17F77) Install (12.75 GB)
Network Disk Image Type

Choose the type of network disk image you would like to create from "Install macOS High Sierra" and click "Next". Customize the disk image using an Automator workflow by clicking "Customize".

- **NetBoot Image**
  Allow Macs to boot over the network from a server-based disk image.

- **NetInstall Image**
  Installs macOS over the network from a hosted disk image.

- **NetRestore Image**
  Restores a volume over the network from an Apple Software Restore disk image.
Software License Agreement

IMPORTANT: BY USING THIS SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE FOLLOWING APPLE TERMS:

A. SOFTWARE LICENSE AGREEMENT FOR macOS High Sierra
B. SOFTWARE LICENSE AGREEMENT FOR SYSTEM IMAGE UTILITY

APPLE INC.
SOFTWARE LICENSE AGREEMENT FOR macOS High Sierra
For use on Apple-branded Systems

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT (“LICENSE”) CAREFULLY BEFORE USING THE APPLE SOFTWARE. BY USING THE APPLE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT INSTALL AND/OR USE THE APPLE SOFTWARE AND, IF PRESENTED WITH THE OPTION TO “AGREE” OR “DISAGREE” TO THE TERMS, CLICK “DISAGREE”. IF YOU ACQUIRED THE APPLE SOFTWARE AS PART OF AN APPLE HARDWARE PURCHASE AND IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, YOU MAY RETURN THE ENTIRE APPLE HARDWARE/SOFTWARE PACKAGE WITHIN THE RETURN PERIOD TO THE APPLE STORE OR AUTHORIZED DISTRIBUTOR WHERE YOU OBTAINED IT FOR A REFUND, SUBJECT TO APPLE’S RETURN POLICY FOUND AT http://www.apple.com/legal/sales_policies/. YOU MUST RETURN THE ENTIRE HARDWARE/SOFTWARE PACKAGE IN ORDER TO OBTAIN A REFUND.

IMPORTANT NOTE: To the extent that this software may be used to reproduce, modify, publish or distribute materials, it is licensed to you only for reproduction, modification, publication and distribution of non-copyrighted materials, materials in which you own the copyright, or materials you are authorized or legally permitted to reproduce, modify, publish or distribute. If you are uncertain about your right to copy, modify, publish or distribute any material, you should contact your legal advisor.
Add Configuration Profiles, Packages, and Post-Install Scripts

No Items

Drag or click (+) to optionally add items.

Customize  Previous  Next
### Add Configuration Profiles, Packages, and Post-Install Scripts

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalAdminAccount.pkg</td>
<td>12 KB</td>
</tr>
<tr>
<td>SuppressSetupAssistant.pkg</td>
<td>8 KB</td>
</tr>
<tr>
<td>munkitools-3.3.1.3537.pkg</td>
<td>3.6 MB</td>
</tr>
<tr>
<td>munki_kickstart.pkg</td>
<td>8 KB</td>
</tr>
</tbody>
</table>
System Configuration

Optionally choose how the system configuration options are applied. If the setting is not defined it will not be applied to the system.

Computer/Hostnames: Choose File...
File contains the computer names and hostnames.

Generate Names: Optional
Generate unique computer names based on string.

ByHost preferences: Match to client after install

Customize Previous Next
Directory Servers

No Servers

Click (+) to optionally add a directory server.
Automated Erase Install

Choose this option to erase a specific target volume and install macOS with no user interaction.

- Erase and install to: Target Volume
Image Settings

Tailor the image settings below prior to creation.

Network Disk: NetInstall of Install macOS High Sierra

Description: NetInstall of macOS 10.13.5 (17F77) Install (12.75 GB).

- Assign a random image index
- Image will be served from multiple servers.
- Assign a specific image index

(1-65535)

Customize
Previous
Next
System Image Utility wants to make changes.
Enter an administrator’s name and password to allow this.

User Name: admin
Password: ********

[Buttons: Cancel, OK]
Image Creation Successful

Your new disk image has been created and is ready to use.
35017071
System Image Utility: NetInstall fails to properly stage multiple additional packages

Created on October 16 2017, 3:38 PM for macOS + SDK

Summary:
A System Image Utility-created NetInstall nbi that contains more than one additional signed packages to install does not properly stage all additional packages for install. This is similar to bug 34979397. After the first reboot after the macOS install environment is set up, the installer stops with the error "The path /System/Installation/Packages/OSInstall.mpkg appears to be missing or damaged." and tells you to restart and try again.

Steps to Reproduce:
Launch System Image Utility.
Select "Install macOS High Sierra.app" as a source.
Choose to make a NetInstall nbi.
Add multiple signed packages to the install.

Expected Results:
macOS High Sierra and additional packages are installed.

Actual Results:
Installer displays the error "This copy of the Install macOS High Sierra application is damaged and can't be used to install macOS" and will not proceed.
When you build your product archives, you should specify a unique product identifier e.g.

```bash
productbuild --distribution Distribution --identifier com.mycompany.pkg.uniqueid --version 1.0 product.pkg
```

You can verify that the product archive contains the product identifier by expanding the created product archive with pkgutil and opening the Distribution file - which should contain a tag e.g.

```xml
<product id="com.mycompany.pkg.uniqueid" version="1.0"/>
```
product id ≠ package identifier
$ pkgutil --expand SuppressSetupAssistant.pkg SuppressSetupAssistant_pkg
$ cat SuppressSetupAssistant_pkg/Distribution

<?xml version="1.0" encoding="utf-8"?>
<installer-gui-script minSpecVersion="1">
  <pkg-ref id="com.github.munki.pkg.SuppressSetupAssistant">
    <bundle-version/>
  </pkg-ref>
  <options customize="never" require-scripts="false"/>
  <choices-outline>
    <line choice="default">
      <line choice="com.github.munki.pkg.SuppressSetupAssistant"/>
    </line>
  </choices-outline>
  <choice id="default"/>
  <choice id="com.github.munki.pkg.SuppressSetupAssistant" visible="false">
    <pkg-ref id="com.github.munki.pkg.SuppressSetupAssistant"/>
  </choice>
  <pkg-ref id="com.github.munki.pkg.SuppressSetupAssistant" version="1.0" onConclusion="none" installKBytes="3">#SuppressSetupAssistant.pkg</pkg-ref>
  <product id="com.github.munki.pkg.SuppressSetupAssistant" version="1.0"/>
</installer-gui-script>
productbuild --identifier com.myorg.uniqueid --package component.pkg product.pkg
pkgutil --expand some.pkg some_pkg
vi some_pkg/Distribution
rm some.pkg
pkgutil --flatten some_pkg some.pkg
35018765
NetInstall: RestartAction on additional packages is ignored
Created on October 16 2017, 4:45 PM for macOS + SDK

Summary:
Additional package's RestartAction is ignored when the package is installed as part of a NetInstall workflow that also installs macOS

Steps to Reproduce:
Launch System Image Utility.
Select "Install macOS High Sierra.app" as a source.
Choose to make a NetInstall nbl.
Add a signed package that requires a restart to the install.
Copy the resulting nbl to a NetBoot server
Boot a client from the nbl and Install macOS.

Expected Results:
macOS should be installed, the additional package should be installed, and the machine should restart.

Actual Results:
macOS is installed and the additional package is installed, but the machine is not restarted.

The package I used for this contains some LaunchDaemons that would normally be loaded...
A workaround would be to create a package script that calls ‘launchctl load’.
NetInstall might work for you if:

• You need to install additional packages:
  • That are distribution-style and contain a “product id” in the Distribution file.
  • That don’t require a restart for full functionality
• The UI is acceptable (no real automation, no menu of choices)
• You have or can set up a NetBoot server
Alternatives?
createinstallmedia
How to create a bootable installer for macOS

You can use an external drive or secondary volume as a startup disk from which to install the Mac operating system.

These advanced steps are primarily for system administrators and others who are familiar with the command line. You don't need a bootable installer to install macOS, but it can be useful when you want to install macOS on multiple computers without downloading the installer each time.

Download macOS from the App Store

1. Download macOS from the App Store, using a Mac that is compatible with the macOS you’re downloading.

To download High Sierra for use with a bootable installer, be sure to download from a Mac that is currently using High Sierra, Sierra 10.12.5 or later, or El Capitan 10.11.6. Enterprise administrators, please download from Apple, not a locally hosted software-update server.

2. When the macOS installer opens, quit it without continuing installation.
3. Find the installer in your Applications folder as a single "Install" file, such as Install macOS High Sierra.

Use the 'createinstallmedia' command in Terminal

1. After downloading the installer, connect the USB flash drive or other volume that will be used as the bootable installer. Make sure that it has at least 12GB of available storage.
2. Open Terminal, which is in the Utilities folder of your Applications folder.
3. Type or paste one of the following commands in Terminal. These assume that the installer is still in your Applications folder, and MyVolume is the name of the USB flash drive or other volume that will be used as the bootable installer:

   ```bash
   sudo createinstallmedia -u /Applications/installer.app -v MyVolume
   ```
$ cd /Applications/Install\ macOS\ High\ Sierra.app/Contents/Resources/
$ ./createinstallmedia --help
Usage: createinstallmedia --volume <path to volume to convert>

Arguments
--volume, A path to a volume that can be unmounted and erased to create the install media.
--applicationpath, A path to copy of the OS installer application to create the bootable media from.
--nointeraction, Erase the disk pointed to by volume without prompting for confirmation.

Example: createinstallmedia --volume /Volumes/Untitled

This tool must be run as root.
% sudo ./createinstallmedia --volume /Volumes/Untitled
Ready to start.
To continue we need to erase the volume at /Volumes/Untitled.
If you wish to continue type (Y) then press return: Y
Erasing Disk: 0%... 10%... 20%... 30%...100%...
Copying installer files to disk...
Copy complete.
Making disk bootable...
Copying boot files...
Copy complete.
Done.
bash-3.2$ ls -a1p /Volumes/Install\ macOS\ High\ Sierra
./
../
.DS_Store
.IABootFiles/
.IABootFilesSystemVersion.plist
.IAPhysicalMedia
.Spotlight-V100/
.VolumeIcon.icns
.com.apple.timemachine.donotpresent
.fseventsd/
Install macOS High Sierra.app/
Library/
System/
usr/
createinstallmedia

+ System Image Utility
Install macOS High Sierra
Add Configuration Profiles, Packages, and Post-Install Scripts

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalAdminAccount.pkg</td>
<td>12 KB</td>
</tr>
<tr>
<td>SuppressSetupAssistant.pkg</td>
<td>8 KB</td>
</tr>
<tr>
<td>munkitools-3.1.0.3430.pkg</td>
<td>3.6 MB</td>
</tr>
<tr>
<td>munki_kickstart.pkg</td>
<td>8 KB</td>
</tr>
</tbody>
</table>
macOS Utilities

- **Restore From Time Machine Backup**
  You have a backup of your system that you want to restore.

- **Reinstall macOS**
  Reinstall a new copy of macOS.

- **Get Help Online**
  Browse the Apple Support website to find help for your Mac.

- **Disk Utility**
  Repair or erase a disk using Disk Utility.

Continue
More alternatives
Imagr is an application designed to be run from a NetInstall environment created with tools like AutoNBI. It can replace tools such as DeployStudio for many organizations, and does not require OS X as its server component, just a simple web server.

This is a Python application, so Python will need to be included in your NetInstall set.

This is under regular development. You may experience bugs if you don't follow the magic path. Issues are welcome, but of course pull requests are even better.

Development Mailing List: imagr-dev
Support Mailing List: imagr-discuss
startosinstall Notes

Greg Neagle edited this page on Dec 22, 2017 · 20 revisions

Table of Contents

- Introduction
- TL;DR
- Requirements
- Workflow Component
  - Optional additional_startosinstall_options
  - Additional package notes and Apple bugs
- Alternate workflow with additional packages
- Testing notes
- More details
  - SIP and startosinstall
  - startosinstall and 10.12.6

Introduction

Imagr can install macOS High Sierra (10.13) to a target volume via the new startosinstall workflow component. You may target an empty volume or volume containing data or a volume containing an OS, just as if you install macOS "by hand."
<array>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/Adminaccount.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/SuppressSetupAssistant.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/munkitools.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/munki_kickstart.pkg</string>
  </dict>
  <dict>
    <key>type</key>
    <string>startosinstall</string>
    <key>url</key>
    <string>http://imagr.fake.com/installers/Install%20macOS%20High%20Sierra-10.13.5.dmg</string>
  </dict>
</array>
<array>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/Adminaccount.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/SuppressSetupAssistant.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/munkitools.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/munki_kickstart.pkg</string>
  </dict>
  <dict>
    <key>type</key>
    <string>startosinstall</string>
  </dict>
</array>
<key>components</key>
<array>
    <dict>
        <key>first_boot</key>
        <true/>
        <key>type</key>
        <string>package</string>
        <key>url</key>
        <string>http://imagr.fake.com/pkgs/Adminaccount.pkg</string>
    </dict>
    <dict>
        <key>first_boot</key>
        <true/>
        <key>type</key>
        <string>package</string>
        <key>url</key>
        <string>http://imagr.fake.com/pkgs/SuppressSetupAssistant.pkg</string>
    </dict>
    <dict>
        <key>first_boot</key>
        <true/>
        <key>type</key>
        <string>package</string>
        <key>url</key>
        <string>http://imagr.fake.com/pkgs/munkitools.pkg</string>
    </dict>
    <dict>
        <key>first_boot</key>
        <true/>
        <key>type</key>
        <string>package</string>
        <key>url</key>
        <string>http://imagr.fake.com/pkgs/munki_kickstart.pkg</string>
    </dict>
    <dict>
        <key>type</key>
        <string>startosinstall</string>
    </dict>
</array>
<array>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/SuppressSetupAssistant.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/munkitools.pkg</string>
  </dict>
  <dict>
    <key>first_boot</key>
    <true/>
    <key>type</key>
    <string>package</string>
    <key>url</key>
    <string>http://imagr.fake.com/pkgs/munki_kickstart.pkg</string>
  </dict>
  <dict>
    <key>type</key>
    <string>startosinstall</string>
    <key>url</key>
    <string>http://imagr.fake.com/installers/Install%20macOS%20High%20Sierra-10.13.5.dmg</string>
  </dict>
</array>
• Packages are staged for “first boot” installation
• macOS High Sierra is staged for installation
• Machine reboots
• macOS High Sierra is installed; firmware is updated; etc
• Machine boots into new macOS High Sierra OS
• Additional packages are installed
• Machine reboots (optionally)
Imagr.app + startosinstall

- "NetInstall"-style NetBoot
- Specially-crafted external boot volume:
  - createbootvolumefromautonbi.py

https://github.com/munki/macadmin-scripts/blob/master/createbootvolfromautonbi.py
bash-3.2$ sudo ./createbootvolfromautonbi.py --nbi ~/Desktop/10.13.3_Imagr.nbi --volume /Volumes/Imagr_Disk
bash-3.2$ sudo ./createbootvolfromautonbi.py --nbi ~/Desktop/10.13.3_Imagr.nbi --volume /Volumes/Imagr_Disk
Mounting ~/Desktop/10.13.3_Imagr.nbi/NetInstall.dmg...
Mounting /private/tmp/dmg.9BnQWH/Install macOS High Sierra.app/Contents/SharedSupport/BaseSystem.dmg...
>>> Copying /private/tmp/dmg.5moDNI/System/Library/CoreServices/SystemVersion.plist
copying file ./SystemVersion.plist ...
480 bytes for ./SystemVersion.plist
>>> Copying ~/Desktop/10.13.3_Imagr.nbi/NetInstall.dmg
copying file ./NetInstall.dmg ...
1815093248 bytes for ./NetInstall.dmg
>>> Copying ~/Desktop/10.13.3_Imagr.nbi/i386/PlatformSupport.plist
copying file ./PlatformSupport.plist ...
5042 bytes for ./PlatformSupport.plist
>>> Copying ~/Desktop/10.13.3_Imagr.nbi/i386/boote
 copying file ./boote ...
571960 bytes for ./boote
>>> Copying ~/Desktop/10.13.3_Imagr.nbi/i386/boote
 copying file ./boote ...
571960 bytes for ./boote
>>> Copying ~/Desktop/10.13.3_Imagr.nbi/i386/x86_64/kernelcache
copying file ./kernelcache ...
25647926 bytes for ./kernelcache
bash-3.2$
Select the system you want to use to start up your computer

- Imagr_Disk
  macOS, 10.13.3
- Macintosh HD
  macOS, 10.13.5

You have selected macOS, 10.13.3 on the disk “Imagr_Disk.”

Click to restart this computer in target disk mode
After you restart this computer in target disk mode, you can connect it to another computer using a Thunderbolt or FireWire cable and use it as a hard disk.

Click the lock to prevent further changes.
NetBoot

Booting from external media
Firmware password protection is off.

Turn on a firmware password to prevent this computer from starting up from a different hard disk, CD, or DVD without the password.

Turn On Firmware Password...

Secure Boot

- Full Security
  Ensures that only your current OS, or signed operating system software currently trusted by Apple, can run. This mode requires a network connection at software installation time.

- Medium Security
  Allows any version of signed operating system software ever trusted by Apple to run.

- No Security
  Does not enforce any requirements on the bootable OS.

External Boot

- Disallow booting from external media
  Restricts the ability to boot from any devices such as USB and Thunderbolt drives.

- Allow booting from external media
  Does not restrict the ability to boot from any devices.
Target Disk Mode

Recovery boot
A bare-bones tool to install a set of packages on a target volume.

Bootstrapper

A bare-bones tool to install a set of packages and scripts on a target volume. Typically these would be packages or scripts that "enroll" the machine into your management system; upon reboot these tools would take over and continue the setup and configuration of the machine.
# hdiutil attach http://macbootstrap
/dev/disk3 GUID_partition_scheme
/dev/disk3s1 Apple_HFS /Volumes/bootstrap

# /Volumes/bootstrap/run

*** Welcome to bootstrappr! ***

Available volumes:
  1  Macintosh HD
  2  bootstrap

Install to volume # (1-2): 1

Installing packages to /Volumes/Macintosh HD...
installer: Package name is foo
installer: Installing at base path /Volumes/Macintosh HD
installer: The install was successful.
installer: Package name is bar
installer: Installing at base path /Volumes/Macintosh HD
installer: The install was successful.
installer: Package name is baz
installer: Installing at base path /Volumes/Macintosh HD
installer: The install was successful.
installer: Package name is Munki – Managed software installation for OS X
installer: Installing at base path /Volumes/Macintosh HD
installer: The install was successful.

Packages installed. What now?
  1  Restart
  2  Shut down
  3  Quit

Pick an action # (1-3):
Managed Software Center

Checking for available updates...

Retrieving list of software for this machine...

Stop
DEP+MDM+...
CUSTOM DEP - PART 1: AN INTRODUCTION

THE FUTURE IS BRIGHT - VERY BRIGHT.

6 MINUTE READ
Mar 8, 2017 • Erik Gomez

CATEGORIES: MACOS DEP MDM EMM AIRWATCH JAMF PRO MOBILEIRON

For the better part of six months, macadmins across the world have been worried about the future of macOS. Whether it's the rumblings people have heard about imaging being killed or that DEP/MDM could possibly be the only way to manage macOS devices moving forward, it has definitely made for an interesting time.

What if DEP was the only light touch/no touch method for installing your company required assets? What if this requirement comes with APFS/10.13?

These are the questions I have had for months. In this time, some macadmins have threatened to retire/leave the ecosystem entirely, while others have raised concerns that their companies may not be able to continue managing Apple products if their toolsets are rendered useless. While both of those thoughts are unsettling, today's post is about positivity.

THIS USUALLY BITTER MACADMIN IS ACTUALLY EXCITED.

Why? Well, for a few months now, I've been busy - V E R Y busy. While some people have started to read between the lines, it's time for me to open up on why I'm excited and how I think many of you will also be eager to join me on this wonderful new adventure.
CUSTOM DEP - PART 2: CREATING A CUSTOM PACKAGE, AND DEPLOYING MUNKI

A LITTLE BIT OF WORK GOES A LONG WAY.

9 MINUTE READ
Mar 8, 2017 • Erik Gomez

CATEGORIES: MACOS DEP MDM EMM AIRWATCH MUNKI

Time to get your hands dirty 😊

In Part 2, you will learn how to create a signing certificate for your packages, sign a basic munki package for deployment and finally create a custom munki package with some defaults.

INTRODUCTION

If you are reading this, I am going to make a few assumptions:

1. You are already using munki in production or have used munki in the past.
2. You have deployed a package in some capacity
3. You understand the basics of munki
Last month I outlined an upcoming feature for a major MDM vendor. In this time, I have received a lot of amazing feedback, but there have also been some questions that have come up.

While I still can’t name the mdm vendor (but I will be able to at the end of this month), people have been trying out SimpleMDM’s strategy around this same feature. Victor and Jesse have also been really working on making MicroMDM a production ready tool.

People are excited and beginning to test this, but it seems that some people have hit a couple of roadblocks.

- How do you create a package to deploy munki and install a profile?
- How do you quickly change your package?
- How do you install tool x and tool y at the same time?
- Can InstallApplication only be used with DEP? (Nope)
Configuration Available

Automatically set up accounts, apps, and preferences.

This Mac will be configured automatically by PretendCo

Administrators may restrict access to apps and features, install and remove apps, remotely erase this Mac, and monitor internet traffic.

Back   |   Continue
Hello DEP!
Sit tight. Your computer will be ready shortly.

Welcome to Pinterest! We're glad you're here!

IT is currently setting up on some things on your device. We will let you know as soon as it's ready for you to use.

helpdesk@pinterest.com
#it on Slack

Running DEP Munki...
Munkiing around with DEP

Sun, Apr 24, 2016
4 minutes

In my last post from November I wrote an introduction to Apple's MDM Protocol spec. Apple has shown an innovative approach to enterprise deployment with it's DEP service and MDM protocol. Apple's solution allows for a more flexible deployment for administrators while giving users more control over their devices. Most enterprises already have robust solutions to manage devices in their organizations - especially laptops and desktops. We use Imagr, Munki and Puppet internally to manage our users' machines.

Unfortunately there are not many options for integrating existing solutions with Apple's new services. MDM vendors offer costly solutions and expect you to adopt their whole platform, no matter how poorly it fits in with your environment. To make matter worse, most MDM vendor's out there don't implement the parts of MDM spec for OS X, instead focusing mostly on iOS.

I've been looking for an MDM solution that would integrate with the DEP service and allow an administrator to assign devices to a group/organizational unit. When that device would arrive in the hands of the end user, the MDM server would bootstrap Munki and Puppet. At that point, our existing services would configure the device as needed. Frustrated by the MDM vendor landscape out there, and inspired by the work of others, I decided to write my own.

I began building an open source MDM server that should fit my needs and hopefully others' too. Today, I want to show you what I got so far. It's not yet ready for release but the source is out there if you want to take a look.

micromdmd

micromdmd is an http server written in Go which implements Apple's MDM Protocol specification. It is fully API driven and can be integrated with other existing services in our enterprise. The server itself is a single binary that can run on Linux/Windows and OS X but it's also designed in a way that allow splitting individual components into their own services.
Installation-based deployment workflow
So imaging is really dead, then?
https://github.com/munki/macadmin-scripts/blob/master/installinstallmacos.py
$ sudo ./installinstallmacos.py --compress

Downloaded http://swcdn.apple.com/content/downloads/41/06/091-82771/93ci062qh1hi2crhtc77wsyemse21xhi8f0x/InstallAssistantAuto.smd
Downloaded https://swdist.apple.com/content/downloads/41/06/091-82771/93ci062qh1hi2crhtc77wsyemse21xhi8f0x/091-82771.English.dist
Downloaded http://swcdn.apple.com/content/downloads/49/22/091-94200/hcsld0r5rq4qdlvkhita7f3gquog1veb/InstallAssistantAuto.smd
Downloaded https://swdist.apple.com/content/downloads/49/22/091-94200/hcsld0r5rq4qdlvkhita7f3gquog1veb/091-94200.English.dist
Downloaded http://swcdn.apple.com/content/downloads/29/03/091-94326/45lbgwa82gbgt7zbgeqlausq2t9zxzl8ku/InstallAssistantAuto.smd
Downloaded https://swdist.apple.com/content/downloads/29/03/091-94326/45lbgwa82gbgt7zbgeqlausq2t9zxzl8ku/091-94326.English.dist

<table>
<thead>
<tr>
<th>#</th>
<th>ProductID</th>
<th>Version</th>
<th>Build</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>091-82771</td>
<td>10.13.5</td>
<td>17F66a</td>
<td>Install macOS High Sierra Beta</td>
</tr>
<tr>
<td>2</td>
<td>091-94326</td>
<td>10.13.6</td>
<td>17G65</td>
<td>macOS High Sierra</td>
</tr>
<tr>
<td>3</td>
<td>091-94200</td>
<td>10.14</td>
<td>18A326g</td>
<td>macOS Mojave Beta</td>
</tr>
</tbody>
</table>

Choose a product to download (1-3): 3
Install macOS High Sierra

Install macOS 10.13.6-17G65.dmg
Target Disk Mode
(or alternate boot disk)
sudo asr restore
--source osx-10.13.5-17F77.apfs.dmg
--target /Volumes/Macintosh HD
--erase
tl;dl
Don’t use imaging to upgrade macOS.
Don’t use imaging to try to bring a Mac from an unknown state to a known state.
Maybe use imaging to setup/restore a Mac:

- To the exact same version of macOS
- On the same hardware (maybe)
iMac Pro:

- Secure Boot
- No NetBoot
- No booting from external media out of the box
You might need new setup workflows.
Recommendations
Modularize your workflows.
Apple installer packages are a good candidate for modules.
Implement an MDM for Mac management.
Get your Macs enrolled in DEP.
Make your management software do (almost) all the setup work...
...by installing all those modules (packages).
Things will continue to change.
Get an iMac Pro for testing.
macOS Installation for Apple Administrators

The guide for “post-imaging” deployment of macOS High Sierra.

With the introduction of macOS High Sierra Apple has profoundly changed installing and managing macOS on a large scale. In addition to the default deployment services like MDM and JAMF, Apple has made macOS High Sierra configurable to such an extent that the possibilities are endless.

This book is designed to guide Apple administrators and suggest best practices to deploy macOS High Sierra in education, business, enterprise and other organizations in this new “post-imaging” world.

Scripting OS X @scriptingosx · Jun 1
"macOS Installation" actually broke into the iBooks Store Top 200!

It passed "Harry Potter and the Deathly Hallows" (for the day)

Big thanks to all, the response is utterly overwhelming.
General-Purpose Deployment Imaging is Dead

Use installation-based deployment workflows instead.
Questions?