

# Practical Deployment Techniques

for Small Labs



Anthony Reimer



Certified Technical Coordinator 10.7



**Session Handbook**  
**PSU MacAdmins Conference**  
**May 2, 2012**

# Packaging

## Why Make Packages?

- ▶ *Push out software via ARD (silently)*
- ▶ *Include (almost) anything in automated image building*
- ▶ *Understand what goes where*

## Packaging Methods

These methods work with JAMF Composer:

- ▶ *Snapshot (Before and After)*
- ▶ *Monitor System Events (fsevents)*
- ▶ *Package Manifests*
- ▶ *Specify Files (including Drag and Drop)*

These methods work with Apple's PackageMaker:

- ▶ *Monitor System Events (fsevents)*
- ▶ *Specify Files (including Drag and Drop)*

## Packaging Tips

- ▶ *Build using same users & permissions*
- ▶ *Segregate user-based and system-based files*
- ▶ *Don't repackage a package if you can avoid it*

## Packaging Tools

### Composer

- ▶ JAMF Software, \$80 (Education)
- ▶ [www.jamfsoftware.com/products/composer](http://www.jamfsoftware.com/products/composer)

### PackageMaker

- ▶ Apple, free w/registration
- ▶ [developer.apple.com/downloads](http://developer.apple.com/downloads)  
(look for “Auxiliary Tools for Xcode” — no longer need to download the whole Xcode bundle in Lion)

### AAMEE (Adobe CS packager)

- ▶ Adobe, free
- ▶ [www.adobe.com/products/creativesuite/it.html](http://www.adobe.com/products/creativesuite/it.html)
- ▶ CS6 will require using AAMEE (3.0)

## Packaging Helpers

### Suspicious Package (Quick Look plugin)

- ▶ *Mothers Ruin Software, free*
- ▶ [www.mothersruin.com/software/SuspiciousPackage/](http://www.mothersruin.com/software/SuspiciousPackage/)

### TextWrangler

- ▶ *Bare Bones Software, free*
- ▶ [www.barebones.com](http://www.barebones.com)

### BEdit

- ▶ *Bare Bones Software, \$50*
- ▶ [www.barebones.com](http://www.barebones.com)

### Apple Remote Desktop

- ▶ *Apple, \$80*
- ▶ *Mac App Store*

# Image Creation

## *Image Creation 1.0 vs. 2.0*

For purposes of this discussion, Image Creation 1.0 is the method where you install software onto a “Golden Machine” then customize it as desired, capturing the result on a drive or in a disk image using tools like Disk Utility, Carbon Copy Cloner or DeployStudio.

Image Creation 2.0 is when you build your images from scratch using packages and an OS installer disk. InstaDMG is one tool that can create an image that will replace your old “golden machine” master. Some 2.0 methodologies involve creating a core image and then applying packages after the core image has been deployed to the target machine. InstaDMG can create these images as well, especially when paired with InstaUp2Date, which allows you to have multiple configurations (in catalogue files).

## Image Creation 2.0 Advantages

- ▶ *Image never booted (no cruft)*
- ▶ *Can swap OSes*
- ▶ *Can roll back problem packages*

## InstaDMG+InstaUp2Date Components

- ▶ *InstaDMG shell script*
- ▶ *createUser package*
  - *with password hash script & text/xml setup file*
  - *Can be used standalone (e.g., create a user account via Apple Remote Desktop)*
- ▶ *InstaUp2Date python script*
  - *with checksumming python script*
- ▶ *Catalogue file(s)*
  - *OS, iLife, iWork catalogue files included and kept updated (see Step 1 of Quick Start guide)*
  - *Templates provided*
  - *Can “include” catalogues in other catalogues*

## InstaDMG+InstaUp2Date Quick Start

1. Get/Update InstaDMG & catalogues
  - `svn checkout http://instadmig.googlecode.com/svn/trunk/  
/path/to/local_folder`
2. Get your install disc (image)
  - *Snow Leopard and earlier:*
    - *Make a disk image from installer disc*
    - *Scan Image for Restore*
  - *Lion:*
    - *Get the Lion installer*
    - *Show Package Contents*
    - *Retrieve Contents/SharedSupport/  
InstallESD.dmg*
3. Place disk image in `InstallerFiles/BaseOS/` or `InstallerFiles/InstallerDiscs` (*latter requires usage of InstaUp2Date*)
4. Assemble own packages in `InstallerFiles/InstaUp2DatePackages`

5. Checksum own packages and add lines generated to desired catalogue files
  - `/path/to/AddOns/InstaUp2Date/checksum.py own_package.pkg`
6. Configure InstaDMG based on a catalogue file
  - `[sudo] /path/to/AddOns/InstaUp2Date/InstaUp2Date.py [-p] catalogue_file`
    - *With -p option, next step done automatically (otherwise, omit items in [])*
7. Go back to Terminal and enter:
  - `cd /path/to/local_folder`
  - `./instadmng.bash`
8. Voilà! A disk image that has never been booted appears in OutputFiles folder.



Make your life easier by setting up aliases for these commands by creating this text file and copying it to `~/.profile` — set whatever abbreviation you want to make a meaningful alias for you (*note also the variation in syntax I've used in this version*):

```
alias upcat="svn checkout http://
instadmng.googlecode.com/svn/trunk
/path/to/local_folder"

alias ck="cd /Users/Shared/
InstaCurrent/AddOns/
InstaUp2Date ; ./checksum.py"

alias configdmg="cd /path/to/AddOns/
InstaUp2Date ; ./instaUp2Date.py"

alias mkdmg="cd /path/to/
local_folder ; sudo ./instadmng.bash"

alias testdmg="sudo asr restore
--target /Volumes/test_partition
--erase --noverify --source"
```

Once you have this file in place, then it is as simple as typing this into Terminal to make your image:

```
upcat
```

```
ck AnyPackageThatHasNotBeenChecked
```

```
configdmg catalogue
```

```
mkdmg
```

The extra command in the `.profile` file allows you to copy the new image to a drive or partition that you have setup for this on your testing machine. Since this workflow will produce the same volume name each time (unless you choose otherwise), you can put that right into your alias. Just make sure you have a backup of any partition associated with your test machine when you debug it!

```
testdmg ImageNameOrDate
```

## Online Resources

### InstaDMG

- ▶ [www.afp548.com/forum/index.php?forum=45](http://www.afp548.com/forum/index.php?forum=45)
- ▶ [code.google.com/p/instadmgl/](http://code.google.com/p/instadmgl/)

### InstaUp2Date

- ▶ [www.osxdeployment.com](http://www.osxdeployment.com)
- ▶ [code.google.com/p/instadmgl/wiki/InstaUp2Date](http://code.google.com/p/instadmgl/wiki/InstaUp2Date)

### InstaDMG + InstaUp2Date (Lion)

- ▶ [www.coolestguyplanet.net/create-an-image-of-os-x-lion-10-7-using-instadmgl/](http://www.coolestguyplanet.net/create-an-image-of-os-x-lion-10-7-using-instadmgl/)

# Deployment

## Deployment Components To Consider

### Cloning

- ▶ *Push the master image onto the target Mac*
- ▶ *Requires a way to boot the target Mac, a repository from which the master image can be served, and a tool to copy the master image to the target Mac*

### Customizing

- ▶ *Everything that needs to be done after the image is cloned onto the target Mac to meet the needs of your environment*
- ▶ *Customizing may include:*
  - *Sharing Name*
  - *IP Address*
  - *Additional Software, Updates (packages)*
  - *Copy protection authorization*
  - *Apply locally managed controls*

Many variations and combinations of tools and methods can be used to achieve these goals.

On a fast (gigabit) network, using DeployStudio will get you excellent performance (the IAML payload of 110 GB can be reliably delivered to 1-3 machines in 45 minutes using a Mac mini server). If you have a slow network, you may consider setting up your test machine to be a NetBoot and DeployStudio server and use an inexpensive 8-port switch and some Cat 5E or Cat 6 cable to create a local network for deployment. Otherwise, DeployStudio can also run from a hard drive (use at least FireWire 800).

## Deployment Resources

### DeployStudio

- ▶ *Freeware*
- ▶ [www.deploystudio.com](http://www.deploystudio.com)

### System Image Utility

- ▶ *Part of Server Admin Tools*
  - *Server Admin Tools 10.7.3:*  
[support.apple.com/kb/DL1488](http://support.apple.com/kb/DL1488)
  - *Server Admin Tools 10.6.8:*  
[support.apple.com/kb/DL1403](http://support.apple.com/kb/DL1403)

# User Management

## How Can You Manage Users?

- ▶ *Remove software from the machine/master that you don't want users to run*
- ▶ *Police manually (i.e. observe users visually) — not a robust strategy*
- ▶ *Use commercial tools (e.g., DeepFreeze, Casper)*
  - *Many require a server*
- ▶ *Parental Controls*
  - *Usable only when using Image Creation 1.x techniques or manually added after deployment*
- ▶ *Managed Client for OS X (MCX)*
  - *Enforce Policy Once, Often or Always*
  - *via Workgroup Manager (10.5–10.7), part of Server Admin Tools*
  - *Server required (unless using Local MCX)*
- ▶ *Profile Manager (10.7 only)*
  - *Policies are enforced “Always”*

## Managed Client for OS X (MCX)

Requires Server Admin Tools

- ▶ *Server Admin Tools 10.7.3:*  
[support.apple.com/kb/DL1488](http://support.apple.com/kb/DL1488)
- ▶ *Server Admin Tools 10.6.8:*  
[support.apple.com/kb/DL1403](http://support.apple.com/kb/DL1403)

## Server-based MCX

- ▶ *Policies can be changed on the server for users, groups, devices, device groups*
  - *goes into effect next login or reboot*
- ▶ *Requires that users authenticate to a Mac server*
  - *“Magic Triangle” allows AD integration*

## Local MCX

Local MCX gives the administrator the same fine-grained controls that are available from server-based policy controls without the need for users to have accounts on the server. Workgroup Manager, working in the local domain, can setup the necessary MCX settings. These can then be captured and deployed more widely.



## Local MCX Quick Start

► *Note: These instructions lock down a single local account (or could be applied to a group). Follow Greg Neagle's instructions for more complicated or robust solutions (see Resources at the end of this section).*

1. Install Server Admin Tools
  - *Use the same user accounts as you do on your master*
2. Run Workgroup Manager and log in to the local directory
  - *Server > View Directories (⌘D)*
3. Click on lock near top right to authenticate as an administrator
4. Click on the Preferences tab
5. Select your restricted user from user list
6. For each category, establish settings you want to manage and Apply them
  - *Always (user cannot change)*
  - *Once (initial value that user can change)*
  - *Never (not managed)*

7. Capture the settings for packaging/  
deployment\*\*
  - `/private/var/db/dslocal/nodes/Default/users/manageduser.plist`
8. Push out to machines needing management using Apple Remote Desktop (or include in image)
  - *MCX settings are applied at next login/reboot*

\*\* A quick way to capture settings using Composer (a similar methodology would work with PackageMaker):

1. Have both Composer & Workgroup Manager running
2. Eliminate `/private/var/db/` from the Composer exclusion list (Preferences > Exclusion List)
3. Make a new package (Monitor File System Changes)

4. Apply any change to MCX settings via Workgroup Manager

- *You can turn something on, click apply, then back off, click apply—anything that causes the modification time to change.*

5. End monitoring

This technique works well for capturing serialization information (the exclusion list change is not usually required).

## ***Using K2 (KeyServer) for App Management***

If you have copy-protected applications to deploy like Finale (MakeMusic!), Max (Cycling 74), Peak (BIAS) or Isadora (Troikatronix), you may find that the vendor supports managing the copy protection through K2 (a.k.a. KeyServer), even if it does not say so on their web site. If you own a couple such apps, KeyServer can pay for itself quite quickly by obviating the need to touch each machine to authorize it (and deauthorize it when you copy a new image to the machine). The company making the copy-protected app may supply you with a text file (.lic) with your license parameters to be installed on your KeyServer server, or they may simply enable management by KeyServer and have you set up your own policy settings to ensure compliance.

If you do make the investment, then KeyServer can also be used to lock down apps with its policy

mechanism. This is an excellent way to lock down apps that are difficult to delete outright, such as Mail (a shell of which could be recreated any time Apple releases a software update). Note that this differs from MCX policies in that it applies to all users of a machine.

It has additional abilities, including quitting idle apps when licenses of that app are requested by another station and locking particular apps to particular computers. K2 also has reporting functions that can assist you with license compliance and seeing how well your app management techniques are working.

## Resources

### Enterprise Mac Managed Preferences

- ▶ *Marczak & Neagle, Apress*
- ▶ *Available in print and through Safari Books Online*
- ▶ *Instructions are based on Snow Leopard*

### K2 (KeyServer & KeyAuditor)

- ▶ *Sassafras Software, ~\$25–29/seat (Educational), sold in blocks of 10 seats*
- ▶ *Subscription: 18% of initial license cost per year*
- ▶ *Requires a server  
(Mac, Windows, Linux, Solaris)*
- ▶ *Clients for Mac, Windows, Linux, Solaris*
- ▶ *Administer from Mac or Windows*
- ▶ [www.sassafras.com](http://www.sassafras.com)

## PSU MacAdmins Sessions

Sessions from the 2010, 2011 and 2012 Conferences provide additional information related to the topics discussed here. Slides as well as a link to iTunes U (for recordings of the presentations) are found at:

- ▶ [macadmins.psu.edu/conference/resources](http://macadmins.psu.edu/conference/resources)