Deployment Guide

OS X

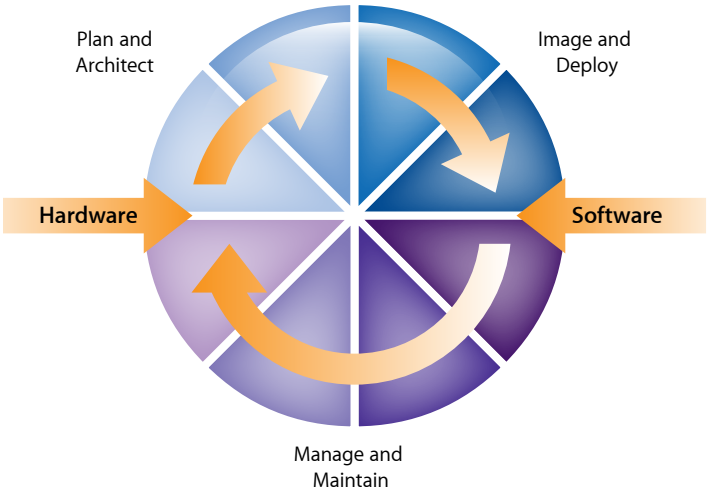
Deployment Guide
March 2013

Contents

Purpose of This Guide

The OS X Deployment Guide is designed to help educational institutions plan, deploy, and maintain Mac systems in 21st-century learning environments. Drawing on over 30 years of leadership experience in educational technology, this guide represents many best practices selected to help IT leaders deliver the best possible tools for teaching and learning.

The system life cycle for educational technology typically takes a year to complete three distinct phases. It starts with a planning phase, when current operations and technologies are evaluated, plans for the next cycle are made, and new or modified architecture is designed. The second phase consists of imaging new systems, reimaging old systems as needed, and deploying the new architecture. The third phase, a year-long operation that overlaps the other activities, involves systems management and maintenance, such as day-to-day client management, patches and upgrades, and other systems management tasks.



The main sections of this guide cover the life-cycle phases: Plan and Architect, Image and Deploy, and Manage and Maintain. All three phases interact to provide a unified approach to hardware and software life-cycle management.

Apple and Educational Technology

Apple has more than three decades of experience in educational technology. Our Apple Classroom of Tomorrow (ACOT) and ACOT2 studies have formed the basis of countless successful deployments. Apple continues to push the envelope with new technology and new ideas. With the growth of 1:1 deployments for student, the introduction of iOS devices such as iPad, and the evolution of educational technology, Apple is helping education institutions achieve their objectives by sharing best practices in technology deployment.

Understanding the Evolution of Educational Technology

Allowing users more flexibility and control over their own devices can greatly enhance the value of technology in the classroom. Schools are making the shift from classic, rigidly controlled labs, where computers are shared by many and used for specific tasks, to a 1:1 deployment model, where users have much more autonomy in their use of systems and devices.

In this modernized environment the focus moves to the user, who can access personal data anywhere and anytime. Methods of mass deployment and mass device management are replaced by personalized management. Users now manage their own devices, regardless of whether the device was provided by the school or owned by the user.

A new model: Personal empowerment
With the evolution of technology comes a new model that puts the end user and their needs at the very core. Personal empowerment allows them to take responsibility for both their systems and personal data. A great example of this would be in an institution where users are given new systems with asset tags attached, and they then become responsible for managing their files, installing applications, and backing up—everyone becomes their own system admin.

In this type of environment, it's key to provide users access to an open wireless network so they can get to the resources needed to complete their work. The deployment of an open filtered, Wi-Fi network in an institution can also provide parents and visitors easy Internet access when on school property. Additionally, this same open network could serve as the primary access method for students and faculty throughout the school day. Access to shared files, school wikis, and other resources can be protected with an account and password, with tighter controls for content used by school administrators and support personnel, such as IT. These secure areas can be protected by a firewall, requiring a VPN for access over Wi-Fi, and can be accessed through hard-wired connections when a user is in a secure office. The faculty could use this secure network for online grade books, student assessment databases, and other noncurricular purposes.

With a reduction in the complexity of the infrastructure, IT staff can gain more time to focus on network performance, collaboration services and user development training.

Modernizing the educational technology space

A modern deployment of educational technology focuses on users and learning curricula. Do the technology and the deployment plans improve or detract from the education mission? It's critical to validate every step in the deployment process to ensure it does not interfere with the primary objectives of your educational institution.

Apple's role as the solutions advisor

Apple's primary role is to help you achieve success by advising you on best practices for deploying transformative technology solutions in education environments. Your Apple account team will provide guidance on solution development for your education mission, driving answers to questions such as:

- How does the technology requirement support the educational mission?
- Why does the requirement exist?

Your account team can provide direction by pointing to successful scenarios and referencing decades of experience with innovative educational technology deployments.

Contacting Apple

To learn more about Apple in education, visit www.apple.com/education, or call 800-800-2775 to speak to an Apple education representative.

Plan

This section will help you examine your current technology environment, compare it with your ideal deployment, and plan your efforts to move in that direction. It starts with an evaluation of your institution's education mission and continues with an assessment of your current infrastructure.

What is your education mission?

What is the mission of your school or institution? In most cases, it's to provide the best possible environment for teaching and learning.

What is the mission of your specific team or group? Why do you do what you do? Does your group's mission support the education mission? How does your technology plan impact, assist, or detract from the education mission?

It's important to define and communicate your education mission before beginning a deployment. Both curriculum and technology leaders should be involved in initial planning meetings around the education mission and defining the technology requirements that support this mission.

Evaluating current infrastructure

All aspects of your current infrastructure—from electrical wiring to available storage capacity on existing file servers—affect the outcome of your deployment plan. For this reason, each of the following areas needs to be carefully examined and evaluated against the goals of the technology plan:

- Electrical
- Networking
- Directory services
- Storage
- Collaboration services
- Training
- Support

Electrical

Start by evaluating your site's electrical grid. Are there sufficient outlets for Mac desktops and notebook power adapters? Can all users plug in their devices at the same time, or can the outlets and power strips accommodate only a few at a time? Is there sufficient amperage to support all of the technology currently in use and planned for use?

Many school sites have been designed for only a few electrical devices per classroom. What would happen if a room full of students needed to charge their notebooks at the same time? For example, recharging 30 MacBook Air notebooks—each of which uses a 45W power adapter (see Apple kbase article)—would require approximately 11 amps [amps = watts/volts or (45x30)/120 = 11.25]. A classroom with a single 15A circuit would be left with very little available capacity for a TV or projector. It would also need its own dedicated circuit, rather than sharing the 15A circuit with another classroom.

A consideration for a large 1:1 deployment is whether users recharge their devices at home, which would reduce the need for large-scale recharging stations onsite. In this case, user training and fallback areas for spot recharging could be valuable solutions.

Networking

What is the current state of your network? What services need to be upgraded? Is your focus on providing access to user data across a wide area network, or is it instead about offering unique collaborative services to specific groups of users? With the shift to 1:1 deployment, your infrastructure may need to bring certain services closer to the local network, as well as reinforce any collaboration services being provided from a central location.

The following sections on core services, wired networks, and Wi-Fi networks suggest discovery questions to help you determine the state of your current network and identify ideas for evolving it.

Core services (ISP, DNS, DHCP)

When evaluating core services, start by diagraming the architecture with the goal of "following the bits." Look for possible data bottlenecks, inefficiencies, and unplanned redundancies. One example is the location(s) of domain name services (DNS). If your Internet service provider (ISP) is the source of all DNS resolutions, every attempt to connect to a named service or URL will need to go all the way upstream to the ISP.

If you operate your own servers on a private network, they should have their own domain names. A local DNS service—perhaps only a few caching DNS servers—should reside within the school building or at a centralized location close to the highest traffic flow. Anything that reduces the time for a client system to resolve Internet lookups is a benefit.

Colocation of servers at a central office can sometimes be a detriment to efficient traffic flow. IT leadership has typically assumed that a high-speed WAN should be more than sufficient for all user traffic. This may have been true in an era when each building had only a few Mac labs. Now, with 1:1 deployment, thousands of users could be accessing central servers over the shared WAN, providing bandwidth of only a few hundred kilobits per second per user. Consider relocating essential services closer to your end users. Centralized management of resources is still possible, but your focus should be on providing end users with the best possible workflow.

Wired networks

Even with the exploding growth of wireless devices in the classroom, a robust wired network remains a key factor in providing a high speed, secure infrastructure.

An important step in evaluating your network is to identify trends in dropped packets and slow responses. By running speed tests at the endpoints of your existing network, you can discover the bandwidth available to the end user. If services are centrally located, these speed tests should also include links across the WAN. Consider running tests from as many endpoints as possible, as well as from all service locations, to create a comprehensive view of network performance.

Overall testing should also include bandwidth speed tests calculated against total users online, to ensure that access to services isn't compromised by slow connections.

Wi-Fi networks

This section can help network administrators with their own deployments or can be used to facilitate discussions with Wi-Fi vendors to ensure an optimal Wi-Fi network design. When preparing the Wi-Fi infrastructure for a Mac (or a mixed Mac/iOS) deployment, consider the following network design factors:

- Required coverage area
- Number and density of devices using the Wi-Fi network
- Types of devices and their Wi-Fi capabilities
- Types and amount of data being transferred
- Security requirements for accessing the wireless network
- Encryption requirements for data passing through the air

Although not exhaustive, this list represents the most relevant factors.

Note: Our discussion focuses on Wi-Fi network design in the United States, and may differ in other countries.

Planning for coverage and density

When providing Wi-Fi coverage for mobile devices, it is essential to plan for the density of devices in a given area. As part of your Wi-Fi network design, consider the expected usage pattern of the Mac systems in your planned deployment.

Most modern, enterprise-class access points are capable of handling up to 50 Wi-Fi clients, although the user experience would likely be disappointing if a single access point had that many devices associated with it. The experience on each device depends on the available wireless bandwidth on the channel and the number of devices sharing the overall bandwidth. As more devices use the same access point, the relative network speed for those devices decreases.

Designing for coverage

Consider the scenario of a district office building with 10 large offices and a conference room on each floor. Spread over two stories are 50 employees equipped with MacBook Pro notebooks as well as iPad and iPhone devices. The notebooks are plugged into Ethernet ports when not mobile, while the iOS devices frequently change locations.

The physical layout of the building encourages informal communication and collaboration. Employees may meet with other employees in conference rooms or in offices. As a result, employees are moving around the building with their iPad and iPhone devices throughout the day, and some employees bring their MacBook Pro notebooks with them. The majority of mobile network access results from checking email, consulting calendars, and browsing the Internet.

In this scenario, Wi-Fi coverage is the highest priority. These mobile users probably won't transfer large amounts of data over the network very often, and the overall density of Wi-Fi devices is somewhat low. The Wi-Fi design could include two or three access points on each floor to provide coverage for the offices, plus one access point in each conference room.

MacBook Pro and iPad both support 802.11n at 5GHz, so the access points could be configured for 802.11n at 5GHz. Because employees use different iPhone devices, a 2.4GHz network must be available. Most modern access points support simultaneous dual frequencies, so support for both 2.4GHz and 5GHz networks should not be a problem. You may also need to enable 802.11b/g to support other mobile devices.

Learn more about support for Wi-Fi standards, including specifications for Apple products, in Appendix A: Wi-Fi Standards and in the Apple support article.

Designing for density

Contrast the district office scenario above with a high school that has 1,000 students and 30 teachers in a two-story building. Every student has been issued a MacBook Air, and every teacher has been issued both a MacBook Pro and an iPad. Each classroom holds approximately 35 students, and classrooms are next to one other. Throughout the day, students conduct research on the Internet, watch curriculum videos, and copy files to and from a file server on the LAN.

The Wi-Fi network design for this scenario is more complex, due to the higher density of Wi-Fi devices. At any given time during the school day, 35 students might be accessing the network, suggesting one access point per classroom. We would also suggest considering multiple access points—the number depending on the density of Wi-Fi devices—for common areas.

Since MacBook Air—the device most commonly used in this school—supports 802.11n at both 2.4GHz and 5GHz, you would want to configure access points throughout the school for 802.11n at 5GHz.

MacBook Air can also benefit from channel bonding, which allows for deployment of more access points without reusing the same channel in nearby locations. However, in this high-density deployment where the majority of devices do not support channel bonding, it may be best to leave this capability disabled. Note that with channel bonding, each access point uses two channels, so fewer channels are available overall.

The above design could be modified slightly if the network must support 802.11b or 802.11g devices. One option is to enable 802.11b/g where dual-band access points are deployed. Another option is to provision one SSID using 802.11n at 5GHz for newer devices and a second SSID at 2.4GHz to support 802.11b and 802.11g devices—taking care not to create too many SSIDs.

In either design scenario, consider avoiding the use of hidden SSIDs, which may delay network association without providing any security benefits. iOS device users tend to change location frequently, and it is more difficult for a Wi-Fi device to rejoin a hidden SSID than a broadcast SSID.

Learn more about Wi-Fi security in Appendix B: Wireless Security.

The above network designs are only examples. The actual design of your environment will vary depending on the unique characteristics of your building(s), user workflows, the specific Wi-Fi devices in use, security considerations, and other factors. To ensure an optimal design, consider collaborating with a Wi-Fi infrastructure provider.

Directory services

Most educational institutions deploy some form of directory services, typically for user and device management. With a focus on personal empowerment, mobile devices and users should now be managed via configuration profiles. Consider, for example, the student in a dormitory or the teacher at home: The device can store policies without constant connectivity to the directory service. At the same time, profile-based management doesn't negate the need for a central repository of users and devices.

When defining a potential deployment, examine current methods and processes to determine necessary upgrades or changes:

- Is your directory architecture centralized or distributed?
- Are all users required to log in to a directory to use a computer or device, or do some users log in to specific services only when needed?

Using the directory service as the starting point, trace the workflow for three categories of end user—school admin, faculty, and student—as they log in to a computer and begin a project. Continue with a "follow-up test," tracing the workflows of the same users as they attempt to reconnect to the previous day's project. Consider the following questions:

- How does that workflow look?
- Are there any steps or procedures that interfere with or block the users' efforts?
- Is any portion of the workflow in conflict with the users' ability to access their data?

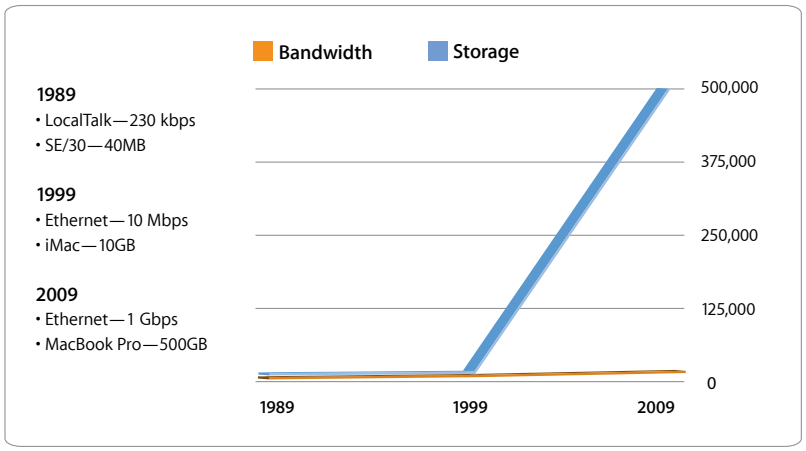
In some cases, there may not be a need to join a user's device to the directory at all. Many users now spend the bulk of their time working locally with documents on their own device and accessing network services to share or archive those files. Some services may require an authenticated logon, but the users opt-in to those services as needed or configured.

Storage

What kind of data storage is provided? The primary uses for data storage in education have traditionally been for network home directories and backup. With the growth of local storage (hard disk space), the evolution of modern applications, and the shift from wired to Wi-Fi networks, the need for network storage has changed. The possibility for users to overwhelm the network while accessing files stored on a network server has become a driving factor in workflow planning.

As the following chart shows, the increase in network bandwidth has not kept up with the increase in storage capacity on local devices. End users are creating documents and projects that exceed the network's capacity for timely transfer. In addition, most modern applications don't allow for the use of network storage, instead requiring local disk access. As a result, user data resides increasingly on local storage, and network storage is relegated to backup or random access.

Network bandwidth vs. user storage



Consider the following questions:

- Was your network storage designed for basic word processing labs, or for a large number of creative users amassing gigabytes of multimedia projects?
- If you are providing backup solutions, are they tailored to fit the needs of end users with more than 100GB of data, including presentations, podcasts, and other media-rich content?

Collaboration services

With the growth of media-rich team projects and with users sharing photos, videos, audio recordings, and other bandwidth-intensive work, it is valuable to explore the need for active, online collaboration services such as wikis, calendars, and iChat. Consider the following questions:

- Are users able to share files in a collaborative project?
- Can faculty members create—and allow student participation through—online blogs and wikis?
- Are students and faculty able to open an online chat to avoid travel for face-to-face meetings?
- Is there a process in place for users to archive completed projects online?
- Does the network design support large-scale collaboration projects?

Training

During the technology planning process, consider exploring two levels of training: user training and support staff training.

With so many advancements in OS X and related applications, user training can help you achieve the best outcomes from your Mac deployment. Valuable end-user training resources include the Apple Professional Development group and other self-help, online, and eBook courseware. Get more information about Apple Professional Development Training.

Documentation
The most current documentation on OS X support and services is available from Apple and from Peachpit Press.

Apple websites

- www.apple.com/education/resources/information-technology.html
- www.apple.com/osx/server/resources/documentation.html

Peachpit Press titles

- *Apple Pro Training Series—OS X Mountain Lion Server Essentials: Using and Supporting OS X Mountain Lion Server*
- *Apple Pro Training Series: OS X Support Essentials*

IT staff training can help develop confidence and flexibility in addressing issues that may arise during your OS X deployment cycle. Apple Professional Training provides a wide range of courses to train your staff on best practices for OS X support. Technical training on the latest system tools and support solutions can also help IT staff stay up to date with technology developments. Get more information about Apple Training for support specialists.

Support

A successful rollout depends on a formalized help desk system, equipped to support different deployment models. For example, in a 1:1 deployment, the support model may benefit from matching that of Apple's Genius Bar, providing both individualized support and a professional development focus.

In a many:1 (shared use) deployment, where the institution has complete control over its Mac systems, the support model may be focused on device repair and reimaging, with limited emphasis on professional development. This would be driven by a need to keep common use systems up and running, versus a focus on teaching the end user more about using and maintaining their own device.

AppleCare

AppleCare products and support are available for institutions of every size.

AppleCare Protection Plan for Mac. Every Mac comes with complimentary telephone technical support for 90 days from the date of purchase, along with a one-year limited warranty. The AppleCare Protection Plan extends the service coverage to two full years from the original date of purchase, and allows users to call Apple technical support experts as often as they like during that period. Convenient service options are available if hardware repair is needed. about AppleCare Protection Plan for Mac.

AppleCare Help Desk Support. This support provides priority access to Apple technical support staff by telephone. It also includes a suite of tools to diagnose and troubleshoot Apple hardware, allowing institutions to manage resources more efficiently, improve response time, and reduce training costs. AppleCare Help Desk Support covers an unlimited number of support incidents for hardware and software diagnosis and troubleshooting, and issue isolation for Apple-based solutions. Learn more about AppleCare Help Desk Support.

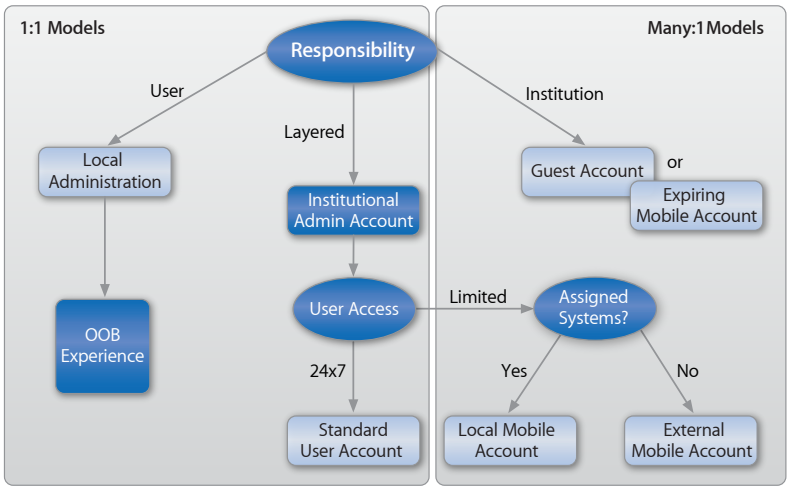
AppleCare OS Support. This combines AppleCare Help Desk Support with enterprise-level incident support, including support for system components, network configuration and administration, integration into heterogeneous environments, professional software applications, web applications and services, and technical issues requiring the use of the command-line tools for resolution. You'll find more information about AppleCare OS Support products.

Apple Professional Services

Apple Professional Services (APS) provides onsite mentoring, assistance, and direct support for deployment. Find out more about how Apple Education Professional Services help you turn technology into educational solutions.

Architect

Once the planning process is complete, you can move on to architecting your next deployment. While clarifying required changes to your infrastructure, it's important to stay aware of how each part of the deployment fits with educational needs. How will end users interact with a Mac system? The following flow chart can help you formulate the best user account types to match desired user scenarios. It begins with the question of responsibility.



Responsibility: Who owns the Mac?

Who will be held responsible for overall, day-to-day management and maintenance of the OS X device—the actual end user, the institution, or both? Which ownership model will provide the best experience for teaching and learning? In reality, the perceived owner may not be the institution that paid for the Mac or the person who images it. In the era of personalized management, the owner may be the person who believes that the Mac is theirs to care for and use. How much control, freedom, and flexibility will individual users have with the devices they use?

This first decision will establish the entire chain of deployment events, as each responsibility model demands a different level of client or systems management. The decision tree includes two primary paths and one layered path for flexibility:

- Users are responsible for the Mac they are using.
- The institution is responsible for the Mac, and users are responsible only for their data.
- The institution has primary responsibility for the Mac, but users may have some level of personal control.

End-user responsibility

If the end user is responsible for the system, the institution provides few, if any, restrictions. IT staff may image the Mac with a standard set of applications, or simply give it an asset tag and turn it over to the user. In the personally empowered model of 1:1 deployment, this assignment of responsibility can motivate the user to gain familiarity and competence with the device.

Even if the educational institution is funding the OS X device, the end user can take primary responsibility for day-to-day maintenance and support of the assigned device. As new technologies emerge and many nontechnical processes are replaced with single-user devices—such as the shift from day planners to online calendars—user responsibility is increasingly common among Higher Education students and faculty, K–12 administrators and IT staff, and other tech aware users.

Assigning primary responsibility to the user provides the most flexibility within the defined constraints of the education mission. The user derives the following benefits:

- The user has maximum flexibility of device and tools.
- Online access and offline access are identical to the user.
- The user perceives the client system as their own computer.

Institutional responsibility

If the institution maintains total responsibility for the device, the goal remains to provide the most consistent user experience possible. In a many:1 model—whether with labs, mobile carts, or kiosks—the Mac will be used by different users. In most cases, IT staff will use its Apple ID to image the Mac with core applications and will apply client management settings to restrict user access to a controlled set of tools.

This model of responsibility has been the norm for the past several decades. The school buys the devices and maintains total control over their deployment and use. The following attributes are key to a successful institutional model:

- Easy access by all users
- The ability for users to access their work and data at all times
- A consistent user experience

Layered responsibility

In a 1:1 deployment, the end user has significant autonomy; while in a many:1 deployment, the user may have control only over personal data. A layered approach combines these two, providing many deployment options.

In the layered model, the institution maintains a core set of controls and/or management of the Mac, but the end user is permitted some flexibility. Each Mac may be bound to a network directory service for user authentication, and possibly to another directory service or Profile Manager for client management settings. This variability enables flexible configurations within the workflows.

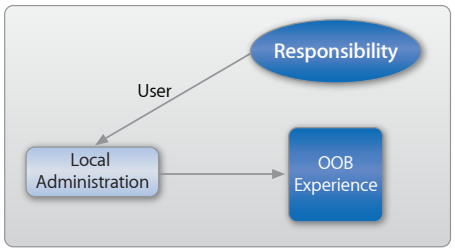
The following attributes are key to a successful layered approach:

- The institution owns all applications, unless the user is also designated as a local administrator.
- Users maintain control of their own data at all times.
- The Mac may or may not be onsite, depending on access requirements.

Workflow-based planning

For each responsibility model, consider a different type of user account. The workflows below describe best practices for each model.

User is responsible: Local admin account



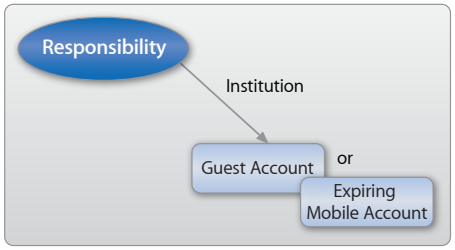
For the 1:1 deployment model, in which the end user considers the Mac a personal device, consider creating an administrator account for each user. This enables the user to edit settings, install applications, and customize the system without negating any infrastructure controls, such as network content filters or asset tracking. Even with an admin account, the school or institution may elect to require VPN access to secure areas, a custom profile for network access, or authenticated access to collaboration services.

End users with an admin account can install applications using an Apple ID. For applications provided by the institution, IT can grant the user a redemption code or a self-service install. In some cases, IT staff may choose to preinstall school-owned applications before turning the Mac over to the user. In this case, they will also need to intervene whenever these applications need updating. For more information on this process, see "Patches and Upgrades" on page 54.

Out-of-box (OOB) experience

The user-responsible model usually results in an easy deployment. The device typically has an asset tag and may be configured using a school-designated image. The device is then turned over to the user to run the setup assistant, where the user creates their own local administrator account. This deployment mimics the experience of consumers who purchase their own system, hence the term "out-of-box experience."

Institution is responsible: Two choices



Guest account

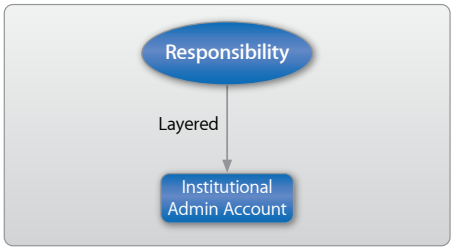
When the institution assumes total responsibility for the Mac, consider creating a guest account at each user login. This approach avoids the need to rely on network directories or special clean-up scripts after each use. If end users need to maintain specific files, you can image the Mac systems with a predefined user home directory template—which could contain, for example, common iPhoto and iTunes content that any user may access during a session.

In this workflow, an end user would log on to a Mac anonymously and would conduct all work locally. The user could copy any personal files to an attached device, such as a USB flash drive, or to a network storage location. At logout, the contents of the guest user's home directory would be permanently deleted. It is critical to alert users that the guest account does not provide any recovery mechanism for user files not copied to another location.

Expiring mobile account

The expiring mobile account is a less extreme approach, providing more flexibility than the guest account. Although the mobile account is network-managed, it is cached locally, so it functions the same as a local user account. IT staff sets an account expiration period, allowing the user to return to the Mac for files and data any time within that period. For example, by setting a three-week expiration period on a shared Mac, the user can come back within three weeks to access personal files. If the user has not logged in to that specific Mac during the three weeks following a session, the mobile account is deleted. The mobile account remains active as long as the user keeps coming back to that specific Mac.

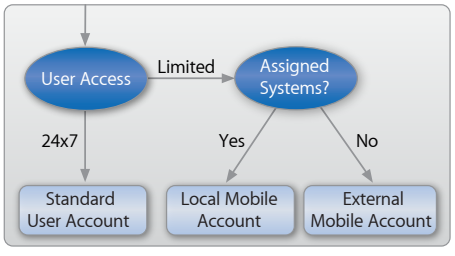
Layered responsibility: Institutional admin account



In the layered model, the institution creates a local administrator account, which may be hidden from the end user's view. This account gives IT staff the ability to perform maintenance and other systems management tasks. With its Apple ID, the institution can install and update core applications on the device. Users are typically assigned a standard nonadministrative account.

Two scenarios for the layered model are a restricted 1:1 deployment and an assigned seat deployment within a lab. In the restricted 1:1 deployment, the end user might not have full control and responsibility for the Mac provided—for example, when the devices must remain at school after-hours or when users are too young to assume responsibility. In these instances, the school or institution creates a local admin account for device management and control, and sets up the assigned user as a nonadministrative account. The specific type of user account is driven by the level of user access to the device.

User access



In a layered responsibility model, the end user will have either full (24x7) or limited (school hours only) access to the device. Limited access may cover a specific portion of the school day or a single class period. The amount of user access drives the last two decision points for the user account type—standard user access or assigned systems (see chart).

Full (24x7) access: Standard user account. When the institution maintains administrative control over an issued Mac but the Mac is always in the hands of the end user. In this model, consider creating a standard user account without administrative privileges. Users have total control over all of their data, which is also maintained locally. This model supports the full creative capability of OS X in the end users' hands, yet allows the institution to deploy a full suite of systems and client management settings and controls.

Note: Users who have a standard (nonadministrative) account will not be able to install applications from the Mac App Store. They can use an Apple ID to set up services such as Mail and iCal, but they would need a local administrator account to install applications on the Mac.

Assigned systems: Seating chart or random access?
In a many:1 model, you will need to decide which scenario will provide the best user experience: a Mac-specific assignment or random access to any Mac. Wherever possible, consider using a specific assignment, because it results in optimal application performance and easiest user access to media.

In scenarios with labs or mobile carts, where users have limited access to Mac systems, the school can create a local administrator account for management purposes and provide a preconfigured set of applications for the end user. Consider exploring scheduling and deployment plans that map out the fewest users per device—for example, rotating a specific mobile cart among a fixed set of classrooms. This would allow you to assign a user to a single notebook in a certain cart. The same approach would work for a computer lab.

Local mobile account. If you decide to assign users a specific system, consider creating non-synced, local mobile accounts for each user. This type of account is network-based, usually from a network directory such as Active Directory, and cached locally on a single computer. The user's credentials come from the network, but the user's home directory, with all of their data, is stored on the device.

The local mobile account benefits the user by providing full access to the creative capabilities of their Mac. The media folder is available to all applications, and the performance of the Mac is the same as if the user were a local account. The only remaining decision has to do with data mobility: Does the user need to transfer any of their information to another system? For more on this issue, see "Data Mobility" on page 49.

This approach is most appropriate when users need consistent access to large amounts of data and large work projects, making mobility a challenge for both the user and the network infrastructure. Examples include high-I/O media projects, such as Apple Final Cut Pro, and large-scale projects that produce many documents.

External mobile account. In situations where the user cannot be assigned a specific Mac, consider using the external mobile account. This approach ensures that the users can maintain their files and data regardless of the device they use.

As with the local mobile account, the external mobile account is cached locally from a network user account, but, in this case, not on the local drive of the device. The account is forced onto an external storage device, such as a USB flash drive, an external hard drive, or an SSD card. This external device becomes, in essence, the user's home directory. The user can plug it into any Mac bound to the network directory and log in using their network credentials. As the user moves from Mac to Mac—or if the external device is formatted for non-Mac systems, from a Mac to a PC and back—all of their media and data will travel with them.

Choosing the right model

Your choice of account models depends on the requirements of your institution. There is no single right approach. For any deployment, you may mix these account models to match a specific mission-oriented scenario.

For example, a faculty member may have an assigned mobile owner/admin Mac, yet need another Mac for research or lab work. You may choose to assign a local mobile account for that second system. The user will have responsibility for transferring critical information between the two systems using a common network location, an external storage device, or a cloud-based storage location.

Another example is a student who switches between a MacBook Air on a cart and an iMac in a lab. The student can be assigned an external mobile account, using their external storage device to preserve their home directory across both systems; or the student may be assigned a local mobile account on both devices, using a common network location to transfer key files as needed.

Image and Deploy

The first step in deploying most systems, including those running OS X Mountain Lion, is to create disk images. OS X Mountain Lion includes robust imaging tools that can be used on their own, or in conjunction with third-party tools, to create deployment images.

IT administrators can choose among multiple imaging methodologies. A traditional monolithic system imaging approach may be best for small proof-of-concept deployments, allowing rapid deployment for user testing. However, most production environments will benefit from the power of programmatic, or modular, image creation workflows that can rapidly deploy systems en masse.

Imaging guidelines

There are two primary methods for imaging a Mac. The first is to perform a clean install, totally erasing the device and placing a full image on its hard drive. The second method is to take a Mac from the box and install additional items to its out-of-box (OOB) configuration.

Clean install: The "nuke and pave" concept
When a Mac deployment plan involves many custom settings and applications, consider building a complete image from scratch. By totally erasing the Mac and imaging a customized system, you can configure it exactly as your institution requires.

The clean install—also referred to as a "pristine condition" system—can be used to restore older systems to a common configuration. This avoids bringing along out-of-date applications, preferences, settings, and other files that accumulate over time. As always, you must take into account legal considerations to ensure that all operating system and application licenses are current.

Clone install: An alternative nuke and pave concept
If the clean install, or imaging, method doesn't fit your requirements, an alternative is to work from a "known good system" for an imaging master. In this case, you would configure a new Mac through the Setup Assistant, create a local admin account, add needed applications, and set appropriate preferences. The configured Mac would then be captured intact as a master image to be cloned onto other Mac systems. The benefit of this method of imaging is that all the subsequent Mac systems imaged will be exactly the same as the master Mac. The downside to this workflow is that you must always use the same make and model of Mac for the entire process, and the possibility of a preference setting that will not behave when cloned between two different Mac models may exist.

Layered install: OOB plus configuration settings
In the case of a large 1:1 deployment, or when only a few applications need to be added to a new Mac, a layered installation may be the better option. You can take the Mac out of the box and install the latest operating system upgrades, the newest versions of applications, and any necessary drivers on top of the factory-imaged system. In an owner/admin situation, in which the user has ownership responsibility, the new items are layered onto the factory configuration—allowing the Setup Assistant to still run at first boot for the end user.

Apple IDs

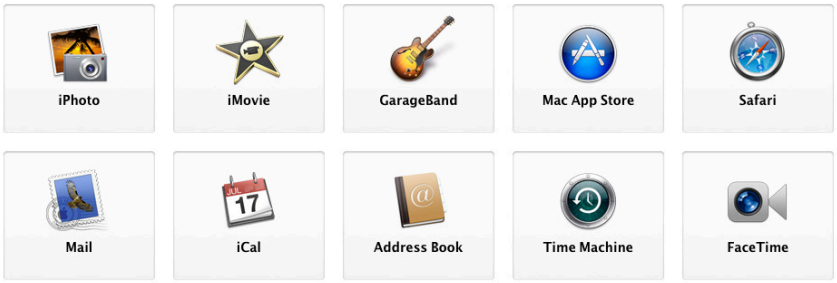
An Apple ID is the single login for most Apple services, including iCloud for content storage, the Mac App Store for application downloads, and the iTunes Store for purchasing songs, movies, and TV shows.

Each Apple ID must be created using a unique email address. Account design varies depending on the deployment strategy. For example, institutions may choose to create iTunes Store accounts without a credit card.

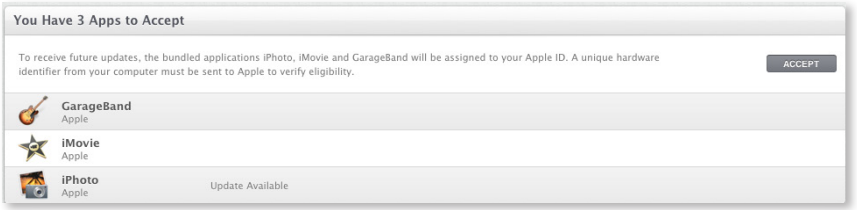
Learn more about Apple ID at www.apple.com/support/appleid.

In-box application issues

A brand-new Mac comes with the latest version of OS X, based on the date the hardware was approved for manufacturing. It also comes with many applications preinstalled (see below).



Of these applications, iPhoto, iMovie, and GarageBand are also considered Mac App Store applications. When you purchase a new Mac, you will need to adopt these three applications, assigning them to the appropriate Apple ID. (Mac customers can update older Mac models to the newest operating system, and then to the latest versions of these three applications through the Mac App Store.)



In a 1:1 deployment, with the end user as owner and admin, you can allow the user to adopt the applications at setup. This assumes that the end user will take responsibility for the Mac over its life cycle; when the unit is past its lease period or due for retirement, the institution will need to replace or retire the applications. In this case, you would treat applications as consumables and account for them in the same way as you treat toner cartridges. If the policies of your institution do not allow for this autonomy, you can use the institution-owned deployment model, as described in the next paragraph. This would not interfere with the end user's ability to install their own applications as the owner and admin of a specific Mac.

If the deployment follows the layered or institution-owned model, you will need to use the institution's Apple ID to adopt these applications. Consider capturing a clean set of the applications on a master Mac, then copying the applications to the other Mac systems. This approach will lock those applications to the institution instead of locking them to an end user.

Mac App Store application installs

Applications purchased from the Mac App Store require either the institution's or an individual user's Apple ID. Applications bought with an individual's Apple ID cannot be recovered or reassigned to the institution's Apple ID. For this reason, consider thinking through your plan for application ownership. Follow the guidelines in the deployment flowchart (see "Responsibility: Who owns the Mac?" on page 14) to determine the most appropriate ownership model.

In a 1:1 deployment, consider providing the user with redemption codes for applications that the school or institution considers mission-critical. These applications and their codes are treated as consumables, just like paper and toner cartridges. While the institution owns the Mac, the applications installed become the property of the end user.

If the school or institution maintains complete ownership and control of the Mac, any purchases from the Mac App Store will use the institution's Apple ID(s). In this scenario, the school will probably use volume licensing (U.S. education only) to purchase redemption codes for Apple-branded applications. For a mass deployment, you need to redeem only one code per application; the other codes will be marked as assigned in the tracking information. You can use the redeemed application download as the master for imaging all your other Mac systems. License management processes will help you keep track of the applications, codes, and associated Mac systems. (See "License Management" on page 54.)

Note: Volume licensing for third-party applications is unique for each vendor, and is not covered in this document.

Updates of institution-controlled applications are handled through IT systems management solutions, discussed later in this guide. Updating an individual user's applications is the responsibility of the end user. For an end user to install applications from the Mac App Store, they must be a local administrator of the receiving Mac.

Media-based application installs

Modern applications, whether purchased on CD/DVD or by electronic download, can be packaged and deployed using the same methods as for Mac App Store purchases, with the additional burden of license tracking. In this instance, consider avoiding setting up an application with Mac-specific serial numbers and instead request that the vendor provide a site license or group serial number for the number of items purchased. (See "License Management" on page 54.)

If the application source requires a drag-and-drop installation or a custom installer, you may have to convert the install to a package using the processes discussed in the next section. In the case of a mass deployment, this will save you from having to manually drag the application onto several hundred Mac systems.

Applications and packages

Before adding applications to a deployment image, you need to turn them into packages. This section covers the different types of applications and how to properly package them.

Note: Although many tools are available for installing applications, this guide focuses on the Apple and third-party tools that are supported by AppleCare or by the company that released them.

Using packages to set up application installs

The use of packages greatly facilitates application installation and management. Most application installers place files on an operating system. A package is a file, or a bundle of files, with a .pkg extension. The package contains an archive of files to install, scripts that perform specified actions (that can run before or after the archive of files are moved to their destination), and information about how the operating system should interpret the installer. A package can also include licensing documents and other information, as needed.

Packages have a number of uses related to installing and managing software. For example, application developers often use packages to build installers for their software. Apple uses packages to provide system or application upgrades via Software Update. Administrators often use packages to deploy small changes to client systems, such as joining to a directory service.

A meta-package, which has an .mpkg file extension, is a set of packages distributed in one structure. It typically provides a list of checkboxes for selecting packages or components of a larger installation framework to be installed.

Many application installers come bundled as standard Apple packages. In situations where an application installer is already a package, you may not need to build your own packages. Vendors that distribute packages often have a process for preparing a package for mass deployment, such as instructions on embedding license keys. In this situation, consider contacting the vendor directly to save valuable time, minimize the amount of user input required to install a package, and avoid unintended consequences.

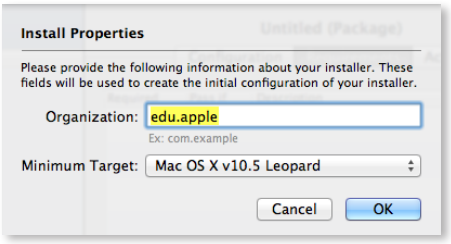
Creating installers for different operating systems is a similar process. Any member of your team who is already trained in creating installers for Microsoft Windows (.msi or .mst installers) or for Linux will quickly grasp the concepts needed to build packages in OS X.

PackageMaker

OS X Mountain Lion includes PackageMaker, a tool for creating installer packages, as part of the OS X Lion Developer Tools, available in the set named "Auxiliary Tools for Xcode" from the Apple Developer website.

Creating an installer package for iWork applications
This example describes the process for creating an installer for the iWork applications—Keynote, Pages, and Numbers.

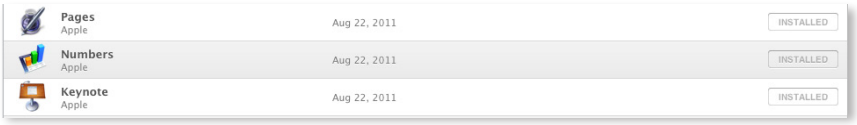
When you launch PackageMaker, you are prompted for the name of the organization that your packages will use and the minimum version of OS X.



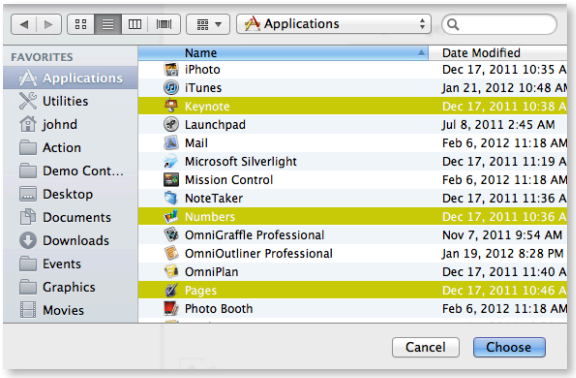
In the Organization field, enter your institution name in reverse domain name order; for example, com.apple for packages developed by Apple or com.microsoft for packages developed by Microsoft. In this example, we use edu.apple.

Using the Minimum Target menu, choose the minimum version of OS X required by clients to run the package. Packages created for OS X v10.5 Leopard are compatible with OS X Snow Leopard v10.6, OS X Lion v10.7, and subsequent versions of OS X. Click OK when done.

Go to the Mac App Store and, using your institution's Apple ID, install the iWork suite to your Applications folder.

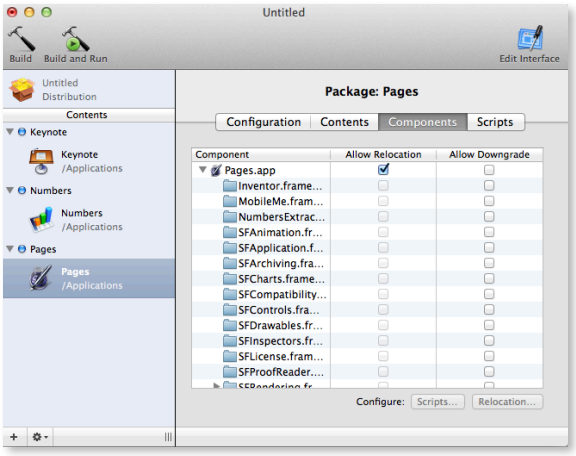


Back in PackageMaker, click the add (+) button in the lower-left corner of the window. Select the files and applications to be included in your package—for this example, Keynote, Numbers, and Pages.

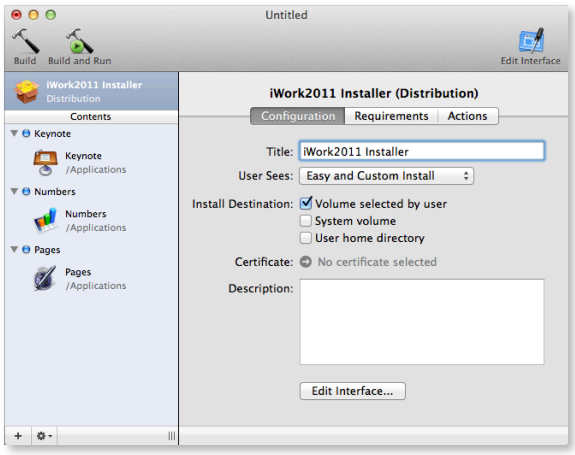


Click the Contents button. Verify that all the files you want to deploy are selected in the Contents pane.

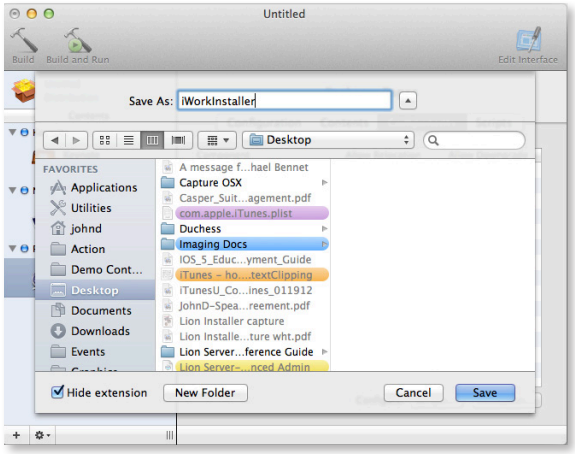
You will see that Pages.app is actually a folder containing a nested hierarchy of files. This is because the Pages.app file, as with a package file, is really a self-contained application bundle. While it is possible to customize application bundles, it is often unwise. Select Untitled Distribution in the upper left of the window. You must give your package a name before creating the installer.



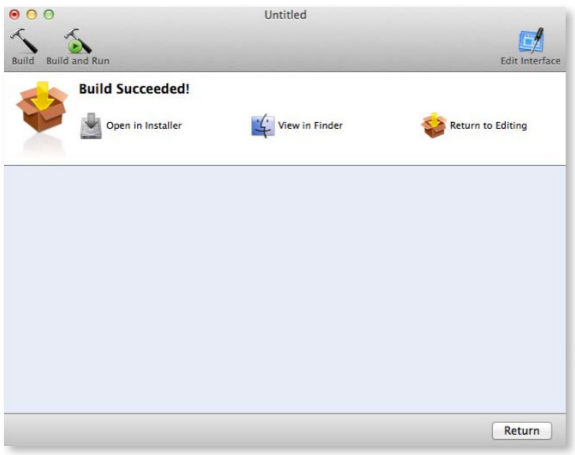
Click "Build and Run" in the upper-left corner of the window.



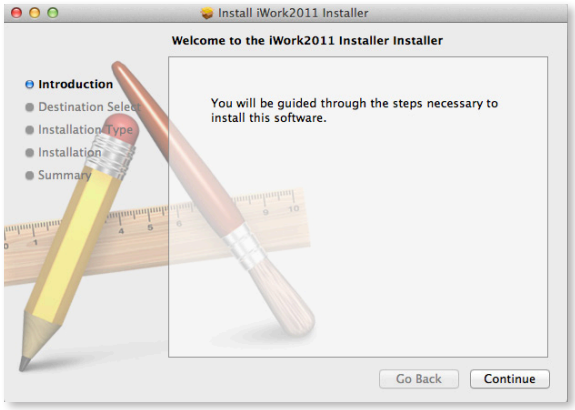
A Save dialog appears. In the Save As field, input the package name (in this example, iWorkInstaller) and choose a destination for the new package. Click Save.



Once the package is built, the installation begins automatically.



You can test your package and, if necessary, use the Return button to further customize the installation. Customizations can include adding unique artwork, a license agreement, or special instructions. For this example, however, the process of building an installer package for the iWork applications is complete.



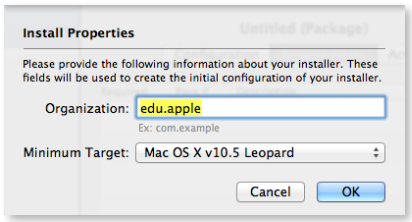
Creating a snapshot package

This section describes how to use PackageMaker to create a package from a snapshot. In this example, the package will contain the Firefox web browser. We will create the package based on changes to the file system during the snapshot process.

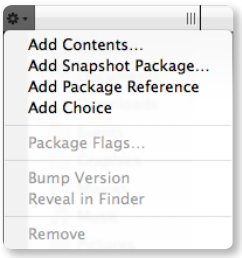
Download the Firefox image to your Mac from www.mozillafirefox.us/mac.html. Open the Firefox image file so you can see its installer.

Launch PackageMaker. When first opened, PackageMaker asks for the name of the institution and the minimum target for the package. For this example, we will use edu.apple in the Organization field and set the Minimum Target as Mac OS X v10.5 Leopard, which limits the package to running on OS X v10.5 or later.

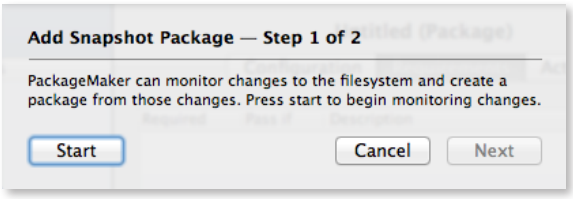
Click OK to start the configuration.



In the Untitled Package window, choose Add Snapshot Package from the Project menu.



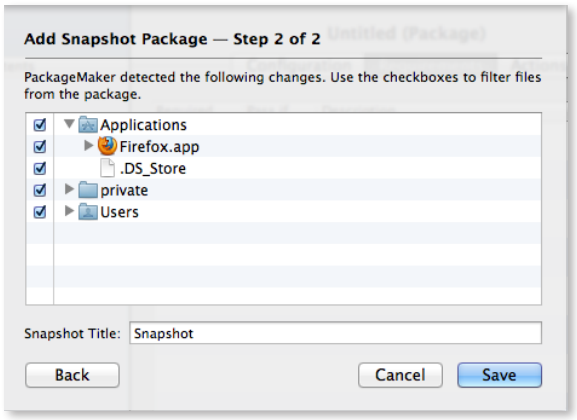
The Add Snapshot Package dialog opens. PackageMaker will create a package based on changes to the file system between the time you click the Start button and the time you click the Next button. To begin, click Start in the Step 1 dialog.



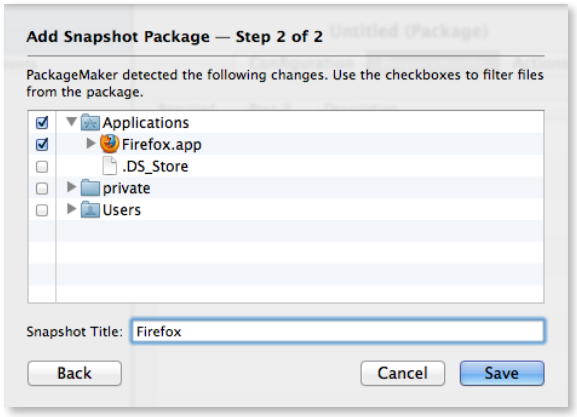
The Start button changes to a Stop button and the cursor spins. Now, install Firefox by dragging the icon onto the Application folder. When the copy is complete, select the PackageMaker dialog and click the Stop button.

Click Next.

A list of files that have been altered or created since you clicked the Start button appears.

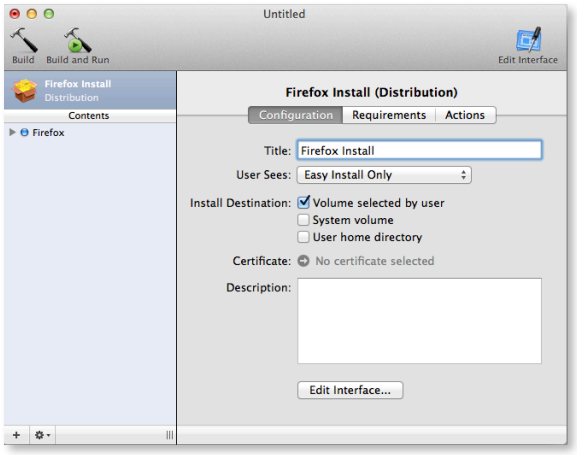


For advanced installs, you can review each file that will be part of the package and remove any that are not necessary. In this case, we want to keep only the Firefox application, so all other items can be unchecked. Change the Snapshot Title to Firefox and select Save.



You will see the snapshot Firefox listed as part of the package. Click the name of the package, then click Configuration.

Provide a package title—in this case, Firefox Install—in the Title field. Note that the text next to the PackageMaker icon has changed from Package to Distribution.

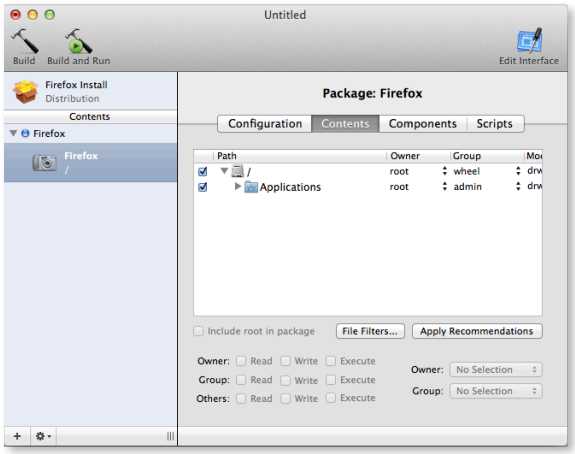


Third-party tools for creating packages
A number of third-party products have compelling features for environments that PackageMaker does not accommodate. These products include the following:

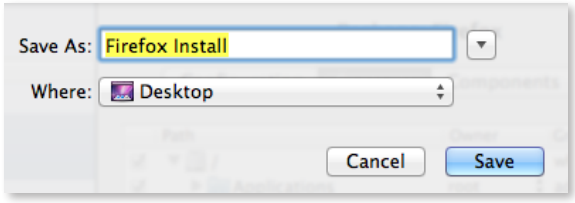
- Composer from JAMF Software
  Use Composer to inspect a computer and create a package of each application that has been installed on that system. Composer offers a smooth transition from monolithic to package-based imaging environments.
- Iceberg and Packages
  Provided under the BSD license, Iceberg and Packages are similar to PackageMaker, but provide additional options for implementing pre- and post-flight scripts, as well as features specifically for meta-package management.
- InstallEase from Absolute Software
  This simple snapshot-based package-generation tool lets you create installer packages with minimal effort.

Click the triangle next to the Firefox item in the left portion of the window (the camera icon indicates it is a snapshot package). The Destination field for the package is displayed. Leave this field as is. The files will be placed in their correct directories in relation to the root of the hard drive, based on where they were on the system that created the snapshot (known as the faux root).

If you change this directory, subfolders will be created in relation to the original hierarchy.



Click "Build and Run" in the upper-left corner of the window. In the Save dialog that appears, enter the name in the Save As field and choose a location for the package.



Test your new snapshot package prior to deployment.

Imaging best practices

Imaging is a critical function of systems management. Major problems can occur if you have not prepared the Mac systems to be used as master images or if you have not collected the proper components for the imaging workflow.

An important step in imaging is the transfer process: how quickly the bits can be transferred from the source to the client. With the introduction of Thunderbolt and MacBook Air, an evolution in imaging practices has taken place. Thunderbolt provides a significantly faster method for transferring an image from a source to the client than network imaging methods. This section discusses several workflows that incorporate Thunderbolt imaging.

The following guidelines cover a set of best practices to assist in a successful imaging operation and deployment.

Preparing systems for imaging

A computer used for imaging should be pristine. It should contain all the files you want to deploy, but no system history, nor any machine-specific data. To build an image like this, it would be wise to remove machine-specific data and any user-specific account information during setup.

Once an image has been deployed, certain tasks need to take place automatically. For example, rebuilding the Local Key Distribution Center (LKDC), joining a directory service, and renaming a computer are easily automated, but will require a minor amount of scripting and/or command-line savvy.

Since ByHost settings are based on the MAC address or UUID of the computer, it's difficult to place certain items in an image and deploy them to local workstations. You can install these items using a post-imaging script or using LoginHooks, so they are run at first login. A better choice in most cases is to establish the items using managed client settings in Profile Manager.

Master client system preparation

If a master Mac will be the image source, perform a clean install of the latest Mac OS, install all relevant applications, and configure the device with the system preferences that will be common to all the deployed systems.

You can find detailed information on creating a master Mac client in the Apple Pro Training Series book, *Mac OS X Deployment v10.6*, by Peachpit Press. Although written for Snow Leopard, this book is also valid for Lion and Mountain Lion deployments.

Customizing the default user template

New user accounts are created with a set of predefined characteristics, including folder hierarchy, preference files, a predefined background, and startup scripts that automatically set up applications. You can customize user accounts using scripts for each setting or based on a customized default user template.

OS X Mountain Lion provides a default user template that populates each newly created account with information in the default user template directory. This means that even after imaging, all new local user accounts will be created with these default settings.

To customize a default user template, start by setting up an account with the required settings and options. Open Terminal from Applications/Utilities.

Note: You need to either log in as root, or prepend sudo to each of these commands.

Use the `cd` command as follows to change your working directory to the /System/Library/User Template/English.lproj folder:

```
cd /System/Library/User\ Template/English.lproj
```

Here you'll find the default files and folders that make up new home directories. Add a file or directory to any of these folders and it is automatically copied to each new user's home folder on the system. An entire account can be set up for this purpose and made into the user template. To do so, first back up the original directory tree to protect against unwanted corruption using the `cp` command as follows:

```
cp -R /System/Library/User\ Template/English.lproj /System/
Library/User\ Template/English.lproj.old
```

After the backup, copy a new directory to the old location. If a local account called Default will be your template user, for example, use the `cd` command again:

```
cp -R /Users/Default /System/Library/User\
Template/English.lproj
```

Whenever a new user is created, all data prepopulated from the default user template will be in the new user's home directory.

Self-removing scripts

Many of the scripts used for mass deployment should be removed from a system after deployment, because the script may contain trusted information—such as a directory services administrative password, a local administrative password, or environment-specific information. By using a shell script—which is capable of removing itself—you can put trusted information into the script without it being exposed inappropriately. However, consider limiting the amount of trusted information added to self-removing scripts, to prevent exposure in the event the script does not complete.

You can use a variety of methods to invoke scripts and to perform tasks with greater control and flexibility. This section looks at how to remove a script once it has been run and how to start a script only after an event has occurred.

The easiest way to remove a script when it is finished running is to add a deletion line at the end. To do so, use the `srm` command, which is a secure version of the `rm` command. For example, to delete a script called `selfdestruct.sh`, use `srm` along with the `$0` option, as follows:

```
/usr/bin/srm $0 selfdestruct.sh
```

Note: Whenever possible, use absolute file paths in scripts.

Once the script has finished running the tasks that come before the line with srm, the script will go ahead and remove itself. If you are familiar with Linux-based systems, you might be tempted to place files that you want to run automatically into the `/etc/rc.local` (which is no longer supported as of OS X Leopard) or in `rc.common` directories. Consider avoiding these locations unless you have a very specific need to do so.

A more flexible method for script removal is `launchd`. By creating a LaunchDaemon or a LaunchAgent, you will be able to include more information in the script and trust that the required script runs, no matter which user logs in to the host. Once the script's tasks have been completed, remove both the script that is invoked by your `launchd` item and the `launchd` item itself.

To create a `launchd` item that starts an application on boot, use the following keys in a property list file (.plist) that is placed in the System/Library/LaunchAgents or System/Library/LaunchDaemons directories. Name this file with a convention that makes sense for your organization—for example, `edu.ourschool.bindscript`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" \
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>BIND</string>
    <key>ProgramArguments</key>
    <array>
      <string>/script_dir/bind.sh</string>
    </array>
    <key>OnDemand</key>
    <false/>
    <key>RunAtLoad</key>
    <true/>
</dict>
</plist>
```

In the above keys, you can set up the string for Label however you want to reference your launchd item in the future. The `ProgramArguments` array will launch a series of scripts—although, in this example, we've used a single shell script located in the /script_dir/bind.sh directory. The `RunAtLoad` key is set to true, which tells the script to launch when the system is booted. At the end of the `bind.sh script` is a line to `srm` the `launchd` item and reboot the host, which self-destructs the script.

To manually launch the script for testing, use the `launchctl` command. To start the script that has been built throughout this example, use the following command:

```
launchctl load -w /System/Library/LaunchDaemons/edu.ourschool.bindscript
```

To stop the script in the event there are problems, substitute `load` with `unload` in the above command. Leveraging an `if/then` statement in a shell script allows you to unload a `launchd` item before deleting it, ensuring it is no longer in use at reboot.

When scripting for first-run events, scripts can be invoked prior to activation of network services. Use the following line early in your script in order to wait until all network services start before proceeding with the rest of the script:

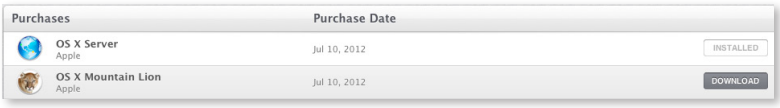
```
ipconfig waitall
```

Locating the correct OS X installer

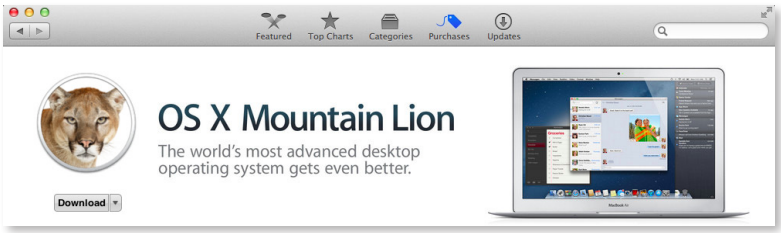
With the introduction of OS X Lion, Apple moved the operating system installer to the Mac App Store, and stopped shipping Mac computers with installer disks. Every time you download the OS X installer from the Mac App Store, it's the latest version of the operating system. The OS X installer is tied to the institution's Apple ID, and the licensing requirements are always in effect, so you no longer need to keep track of the original OS X installer and match it with updates.

Capturing the latest OS X installer image

On a Mac owned by your institution, launch the Mac App Store application. Sign in using an institution Apple ID. Locate the OS X installer under Purchases.

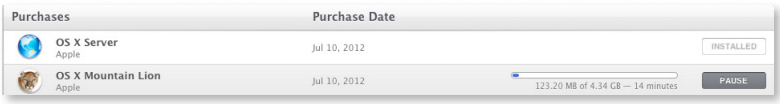


Option-click the OS X installer icon, and you will see the detail page for the OS X installer, in this case, OS X Mountain Lion.



You can see the information on this installer to be sure that it's the current version.

Click the Download button to begin downloading the OS X installer onto your local Mac.



When the download is complete, the installer will launch automatically. Since you don't need to install at this point, you can quit the installation.

Locate the OS X installer in Applications. This installer application is all you will need to begin creating OS X image sets. Some third-party imaging tools may require you to extract the image files from within the installer, but the Apple tools require only that the application be present in the main Applications folder.

Imaging using Disk Utility

An image is a representation of a computer and its related information—including the kernel, file systems, libraries, and programs—at a given time. By contrast, a disk image is a representation of the file system, typically captured while offline to create a complete image of the system. When we refer to an image in this document, it is one of the following:

- A single .dmg file that stores a monolithic representation of a Mac and can be copied in its entirety to other Mac computers, or a creation of packages that make up a modular representation of that .dmg file.
- A Mac system that can be copied in an object-oriented fashion to other Mac computers.

Creating an image of a hard drive and copying it to another hard drive is a basic operation of OS X Mountain Lion. Images can be deployed directly through target disk mode or from one disk to another; or over a network using NetInstall/NetRestore or a third-party product.

Target disk mode
If the Mac you are working with has a FireWire port, target disk mode enables the Mac to act as an external FireWire drive when connected to another Mac with a FireWire port. If the Mac has a Thunderbolt port, it will act as an external Thunderbolt drive when connected to another Mac with a Thunderbolt port.

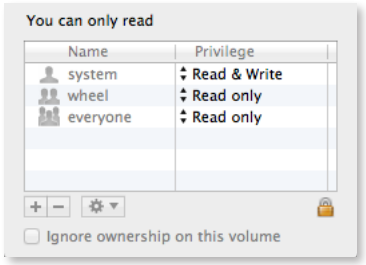
Creating an image using Disk Utility

In this example, we look at how you would use Apple's Disk Utility, located in Applications/Utilities, to create an image of a hard drive.

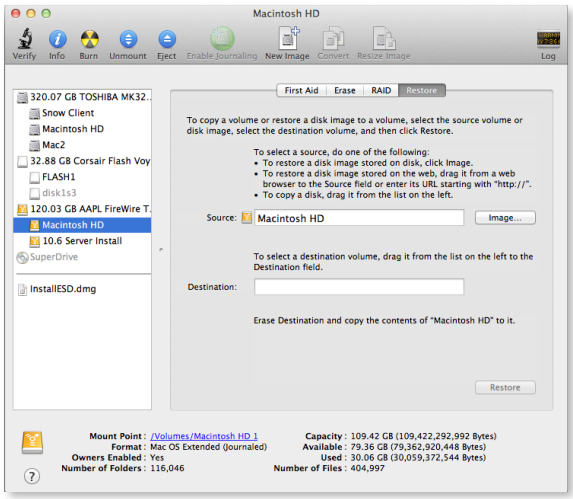
Start by building the perfect system image. First, install the operating system and required software, preferably using Volume License Agreement licensing, and configure the various settings.

Reboot the system into target disk mode by holding down the T key during the boot process.

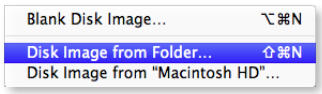
Connect the image source computer to an image creation computer and verify that the hard drive mounts. Select the volume and choose Get Info from the File menu (or press Command-I). Verify that the "Ignore ownership on this volume" checkbox is not selected.



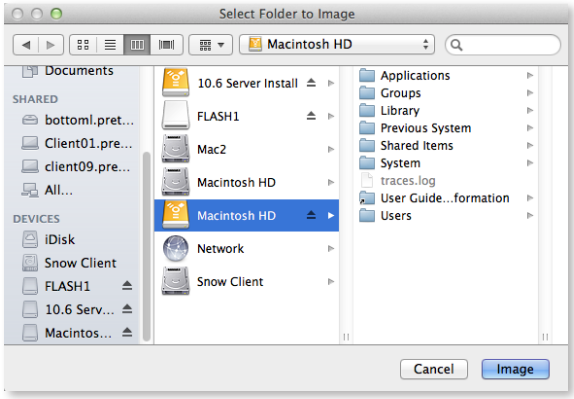
Open Disk Utility.



Choose New from the File menu, and choose "Disk Image from Folder…"



The "Select Folder to Image" pane allows you to choose the volume from which to create the image. Select the name of the hard drive of the prepared client, which you should have already booted into target disk mode. Click the Image button.

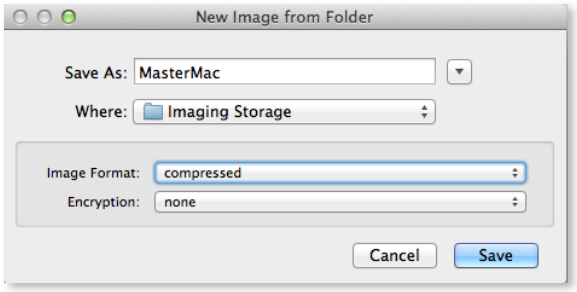


In the "New Image from Folder" dialog, provide a name for the image—in this example, MasterMac.

Use the Where menu to define where on the system to create the image file.

Choose "none" from the Encryption menu.

Click Save to create the image.



Wait for the image to complete. The time depends on the size of the image and the speed of media for both source and destination.

Once the image is complete, unmount and remove the hard drive that was used as the image source.

In Disk Utility, choose "Scan Image for Restore" from the Images menu.

Select the image previously created. This process will reorder the bits and produce an index for insuring that all the right parts of the image get laid down on the target Mac properly.

Creating a disk image from the command line

Apple Software Restore (ASR) enables you to create images from a disk. In this example, we create an image from the command line.

The `hdiutil` command can be used to burn, create, expand, and verify disk images. Here we use the `hdiutil` command to create the image `.dmg` file by invoking the create verb when you run it. Mount a drive called MACOSX that houses an image of a clean OS X Lion installation onto your computer, and create an image of it. Call the image `LionImage` and place it in the Desktop folder on the computer. The following command illustrates a simple way to create the `.dmg` file:

```
hdiutil create -srcfolder /Volumes/MACOSX ~/Desktop/
LionImage.dmg
```

Next, have the `asr` utility scan the image using the following command:

```
asr imagescan --source ~/Desktop/LionImage.dmg
```

In this case, `asr` is used with the `imagescan` verb to calculate the checksums of the contents of the image file and to store them in the image. These checksums will verify that restores occur properly. The `imagescan` verb will also reorder files so that the image can be deployed in a multicast fashion. The `--filechecksum` and `--nostream` options work with the `imagescan` verb to calculate checksums on a per-file basis and bypass reordering of the files, respectively.

Note: By default, Disk Utility creates an image up to 256GB. To create a larger image—up to 512GB in this example—use the following command to set defaults before using Disk Utility or `hdiutil`:

```
defaults write com.frameworks.diskimages \
hfsplus-stretch-parameters -dict \
hfsplus-stretch-threshold 524288 \
hfsplus-stretch-allocation-block-size 4096 \
hfsplus-stretch-allocation-file-size 16777216
```

Imaging using System Image Utility

System Image Utility (SIU) is the primary tool for creating NetBoot images. SIU can create three types of images—NetBoot, NetInstall, and NetRestore—and two workflows (basic and custom). SIU is included in the core OS for Mountain Lion inside the System/Library/CoreServices folder. In Lion, it is part of the Server Admin Tools from Apple Support downloads (support.apple.com/downloads).

This tool can work as part of your imaging or maintenance solution set as follows:

- **NetBoot images** can provide network-accessible repair disks. Instead of carrying a bag of external drives, USB flash drives, and CDs/DVDs, support staff can boot a troublesome Mac from a network image that contains Disk Utility and other key diagnostic and repair tools.
- **NetInstall images** can install OS X Mountain Lion and additional packages over the network. Use a NetInstall image if you plan to upgrade a volume from a previous version of OS X without first erasing it.
- **NetRestore images** can erase the destination volume and perform a block copy of a preinstalled system over the network. NetRestores are typically much faster than NetInstalls.

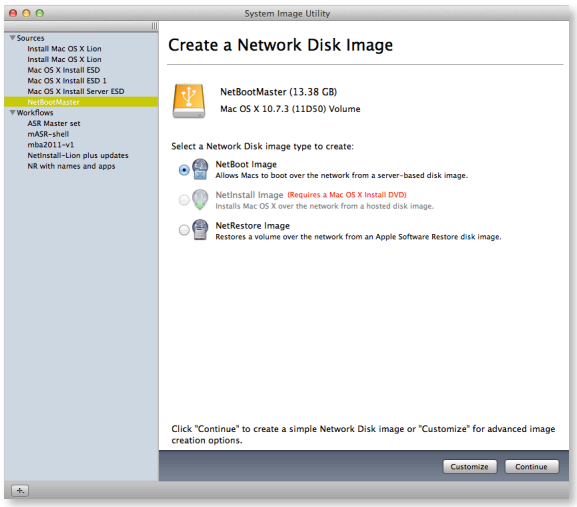
Creating a basic NetBoot image

NetBoot enables you to boot a Mac computer from a disk image of OS X Mountain Lion stored on a computer running OS X Mountain Lion Server. Client computers receive the image across a network via HTTP or NFS.

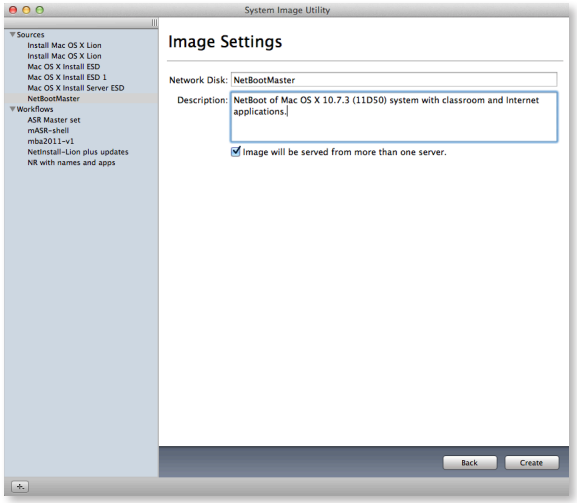
The basic NetBoot image starts with one of two options: an OS X install disk image or a known configured Mac. This second option could be a Mac set up as a repair and diagnostic system with various tools installed, or a classroom system bound to a directory server with a limited set of applications for testing and Internet research. The key is to make sure the Mac is working properly before converting it to a NetBoot image set. You may also install OS X on a partition of an external device and set up that partition as your master Mac system.

Launch System Image Utility from System/Library/CoreServices/System Image Utility. Your source will be the previously downloaded Mountain Lion install application or your preconfigured master Mac system. In this example, we chose a USB-mounted partition configured as a fully functional Mac OS and application set for a classroom.

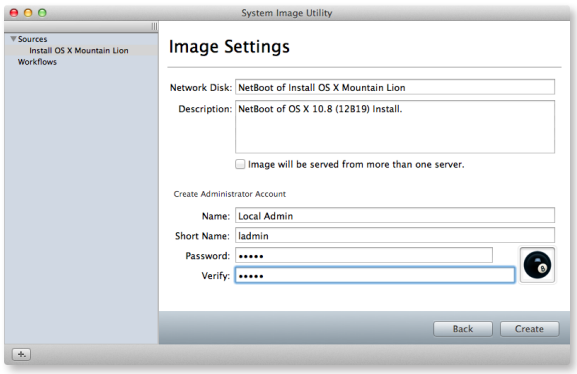
Click Continue.



Type in a name for the image and a description. You can also choose to provide this image from multiple copies of OS X Server, which improve performance when many are using the NetBoot process.



If you used the OS X Mountain Lion install application for the NetBoot source, you would be asked to provide a local administrator's name and password. When the NetBoot image is created, it will be a basic OS X install with a single local administrator account.

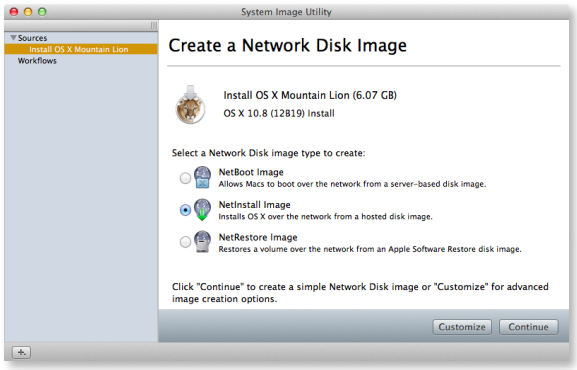


Select Create and choose a location to store the image. If you are using SIU on OS X Server, you should choose the /Library/NetBoot/NetBootSP0 directory.

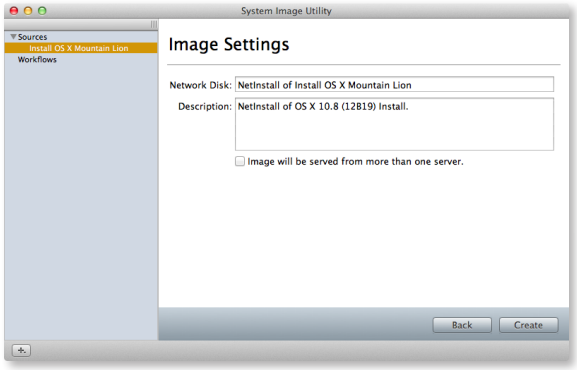
Creating a basic NetInstall image

Download the latest OS X Mountain Lion Install application using the procedure outlined earlier.

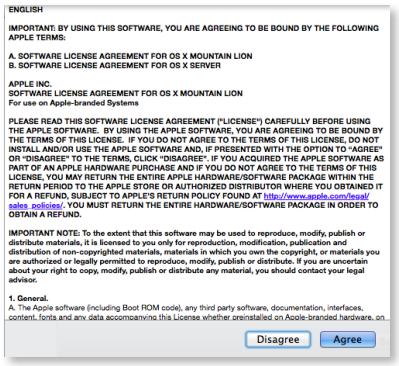
Open System Image Utility from the System/Library/CoreServices folder. In the "Create a Network Disk Image" pane, select NetInstall Image. Click Continue.



In the Image Settings pane, enter a name for the mounted disk in the Network Disk field. Describe the image in the Description field. If more than one OS X Server will house the NetInstall image, click the "Image will be served from more than one server" checkbox. Click the Create button.

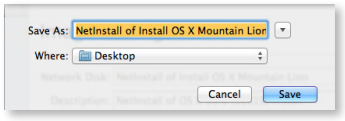


Click Agree in the Software License Agreement pane, if you agree to the license agreement.



Enter a name and choose a destination for the image.

Note: You can change the menu to the Library/NetBoot/NetBootSP0 directory if you have already set up your NetBoot server and are running SIU from the server.



Click the Save button to generate the NetInstall image.

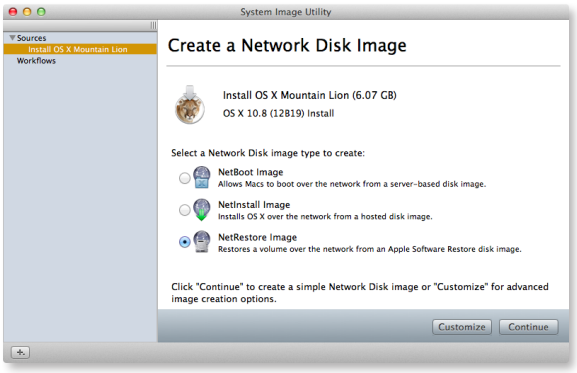
Creating a basic NetRestore image

NetRestore images can erase the destination volume and perform a block copy of a preinstalled system over the network. NetRestore operations are typically much faster than NetInstall.

Along with creating a NetRestore image from a preconfigured volume, you can use SIU to create a NetRestore image of the OS X Mountain Lion Install application. Creating the NetRestore image in this way provides a pristine installation of OS X Mountain Lion.

Download the latest OS X Mountain Lion Install application, or attach an external drive with a partition containing a completely configured OS X system.

Launch SIU from System/Library/CoreServices/System Image Utility. Click the NetRestore Image button and select either your Install ESD image or your prebuilt OS X system.



Click Continue.

When using an install source, type in a name or description for the network disk and fill in the values for the local admin account. If you are using a prebuilt OS X system, you will have already created a local admin account, and this option will not be available.

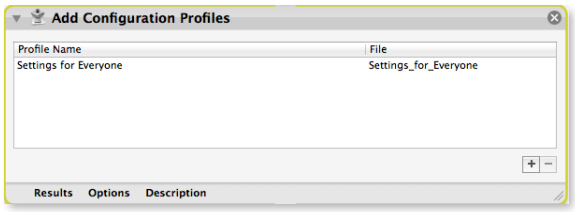
Click Create and choose where you want to store the NetRestore image set. If you are performing these tasks on OS X Server, you should have already set up NetInstall services and will use the Library/NetBoot/NetBootSP0 directory.

Creating custom workflows

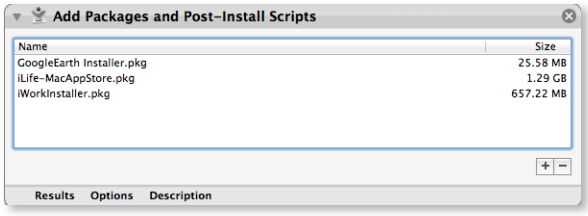
Customization capabilities in System Image Utility add flexibility to your imaging tasks. Using Automator actions, SIU can create uniquely customized network image sets for a wide variety of uses. While a full discussion of possible Automator actions is beyond the scope of this guide, the following section will cover those actions designed specifically for SIU.

SIU Automator actions for customization

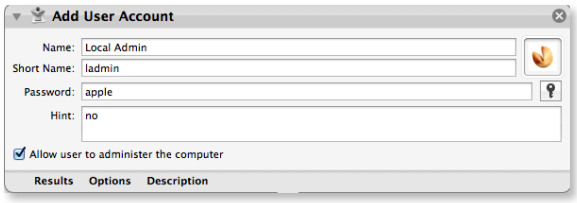
Add Configuration Profiles. This action works with profiles created in Profile Manager. You can add profiles to the image set so that imaged Mac systems start up with complete profile management configurations. Details on profiles and their use in systems management are covered in "Manage and Maintain," which starts on page 46.



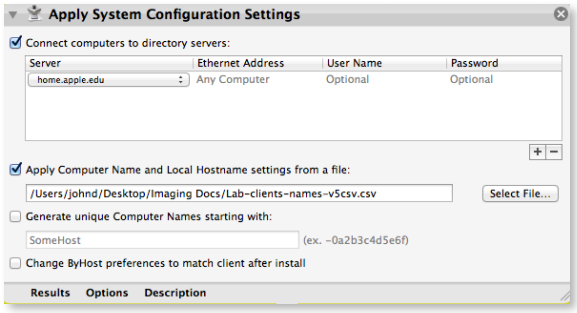
Add Packages and Post-Install Scripts. Following the guidelines in the section on creating packages, you can add .pkg files to your installation. You can also add any number of post-install scripts that perform special operations on an imaged Mac.



Add User Account. This action allows you to add a user account to an image made from an install disk. The user can be an administrative user or a local standard user. If you create a single-user account, consider setting up the local administrator as the user. To add several user accounts, you can plug in this action more than once in the workflow.



Apply System Configuration Settings. This action allows you to configure bindings, generate unique machine names, and change ByHost settings. Bindings can be anonymous or user and password configurations.

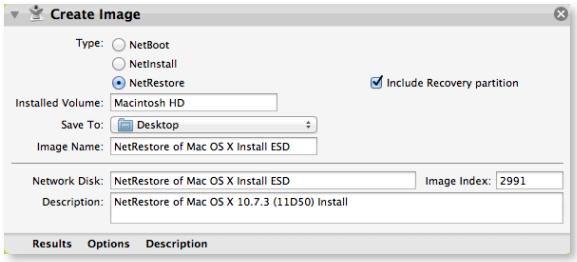


Note: The format for the computer name file is shown below:

macAddress	hostName	computerName	bonjourName
00:1F:F3:4F:F0:AC	-automatic-	lab-imac-24	lab-imac-24
D4:9A:20:C4:48:04	-automatic-	lab-macbook-w1	lab-macbook-w1
00:1F:F3:45:20:FF	-automatic-	lab-macmini-1	lab-macmini-1
00:1F:F3:44:E0:B9	-automatic-	lab-macmini-2	lab-macmini-2
00:23:DF:9D:7D:24	-automatic-	lab-macmini-3	lab-macmini-3
D4:9A:20:F8:F1:26	-automatic-	lab-macmini-4	lab-macmini-4
D4:9A:20:0A:4E:2C	-automatic-	lab-mbp-13	lab-mbp-13
C4:2C:03:3C:E4:4D	-automatic-	lab-mbp-15	lab-mbp-15
28:37:37:1C:8F:12	-automatic-	lab-mba11t-1	lab-mba11t-1
B8:8D:12:00:6C:E2	-automatic-	lab-mba11t-2	lab-mba11t-2

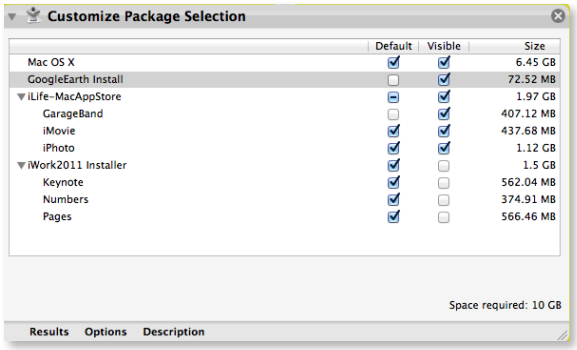
The file can be saved as text or in CSV format. If the headers are included in the file, they will be ignored by SIU. If you prefer to use a different hostName, substitute it for `-automatic-`; otherwise, the hostName will be the same as the computerName. The same option applies to the bonjourName. In all cases, you must enter a name in at least one of the three name fields.

Create Image. One of the required actions is the Create Image action. Always the last to run, this action allows you to name the image set, add a description, choose the location for the image set at creation, and decide if you want to include an OS X Recovery HD with your image process. This also allows you to custom-select the image index number, which is the value used by the NetBoot process to uniquely identify the image set.



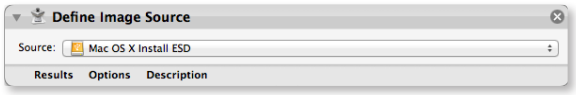
Note: Index numbers from 0 to 4095 are designed for a NetBoot image to be provided from a single OS X NetBoot server. Index numbers from 4096 to 65535 are designed for the same NetBoot image to be provided by two or more NetBoot servers simultaneously—a process that increases performance for large NetBoot deployments.

Customize Package Selection. This action allows you to customize packages you add to an install and to customize the OS X install set. You can also force certain packages to install and designate other packages as optional installs.



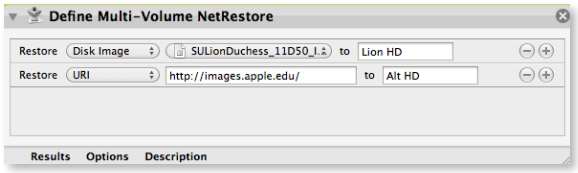
The Default option sets the specific item to be installed unless unchecked by the user during installation. The Visible option enables the user to see the item and choose whether to install it or not. For an automated installation, all items are checked and will be installed.

Define Image Source. This action asks you for the source image for your NetBoot set. It requires an install disk or a nonbooted, preconfigured OS X disk, or partition.

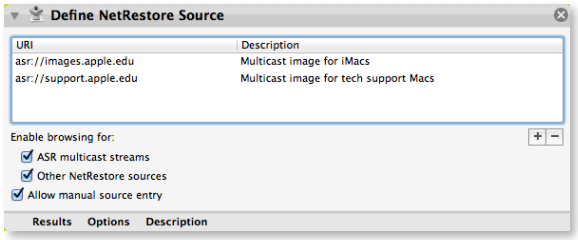


Note: Define Image Source is one of two mandatory starting actions (the other is Define NetRestore Source, see below). One of these must always be the first action in the workflow.

Define Multi-Volume NetRestore. This action allows you to define multiple Apple Software Restore (ASR) sources for NetRestore imaging of a system that has been partitioned into multiple volumes.

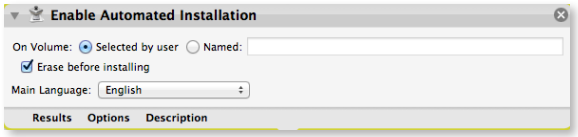


Define NetRestore Source. With this action, you can create a NetRestore shell environment that will restore configured http disk images and multicast ASR streams. It allows you to provide a list of restore sources, scan for restore sources, and manually enter image source URLs.

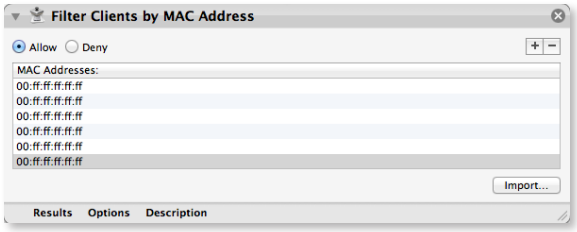


Note: As before, Define NetRestore Source is one of two mandatory starting actions (the other is Define Image Source). One of these must always be the first action in the workflow.

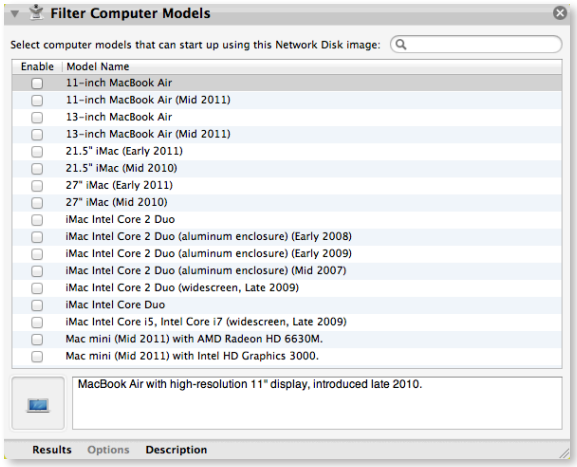
Enable Automated Installation. This action allows you to configure an unattended installation process. You can choose the specific volume to be imaged, whether the target is erased, and the primary language to be used.



Filter Clients by MAC Address. This action allows you to designate a specific set of Mac clients to be imaged using a text file that contains their MAC addresses.

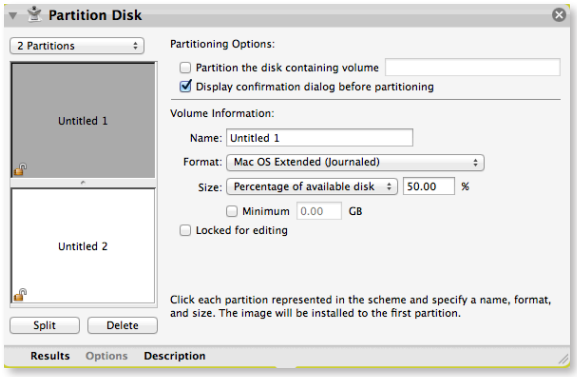


Filter Computer Models. You can select specific Mac models for the install/restore image set.



Note: As of the writing of this guide, 47 distinct models were listed.

Partition Disk. This action allows you to partition the target system before imaging.



Applying custom Automator actions to NetBoot sets

The following table shows which actions apply to a specific imaging set—NetBoot, NetInstall, or NetRestore.

SIU Automator Action	NetBoot	NetInstall	NetRestore
Define Image Source (often required)			
Create Image (required)			
Add Packages and Post-Install Scripts			
Customize Package Selection			
Add User Account		Not Applicable	
Apply System Configuration Settings		Not Applicable	
Enable Automated Installation	Not Applicable		

Some actions work only with an install disk as the source. Other actions work with either an install disk or a preconfigured Mac image.

Thunderbolt and imaging MacBook Air

The ability to rapidly image large numbers of Mac systems for deployment is an important requirement for education system administrators. With the introduction of the 2011 MacBook Air, new hardware features—including all-flash storage and Thunderbolt, Apple's next-generation high-speed I/O technology—challenge the traditional performance and cost advantages associated with network-based restore.

For more information, see the white paper, "Imaging the MacBook Air."

Manage and Maintain

As an IT professional at an educational institution, your role is to design, test, deploy, and manage clients and servers throughout your organization—whether it's a single elementary school or a large university with campuses around the world. At the same time, the technology solutions you deploy must meet the many (sometimes conflicting) requirements of teachers, students, researchers, and administrators in these institutions.

Systems management life cycle

To identify the best possible technology solutions, it's helpful to treat systems management as a continuous process of discovery, adaptation, and maintenance. With new iterations typically occurring every school year, this systems management life cycle involves evaluating current systems, modifying requirements, testing and deploying new systems, and managing those systems as long as they remain in place.

Evaluate

Toward the end of the fall quarter, it's appropriate to determine with teachers and administrators how the entire educational technology ecosystem is functioning.

- Is the current set of hardware and software satisfying faculty and student needs?
- Are there technical issues that need to be addressed?

This portion of the cycle may take several weeks or several months. Once the evaluation is complete, your job is to recommend additions, replacements, and modifications to software and hardware deployments.

Deploy

Now that you've evaluated the technology you have in place, you can develop a technology plan for the next school year. This phase includes testing new sets of hardware, software, and management tools, and modifying them as required. Deployment testing can run from the end of the fall quarter all the way to spring quarter. After the final configurations are in place, you can conduct further testing in a live environment. Many schools begin pilot deployments during spring break and test until the end of the school year, or even through summer break.

Manage

When deployment is complete, the management phase starts and extends throughout the year, overlapping the next iteration of evaluation and deployment. As the cycle progresses, some steps may iterate. For example, images may need to be tested repeatedly, and systems may require occasional updates or repairs.

Defining systems management tasks

OS X deployment in an educational institution requires the following systems management tasks:

Asset management. Tracking institutional purchases, such as tagging Mac systems and iOS devices—also known as keeping track of stuff.

Imaging. Imaging Mac systems with a preferred set of applications and operating system version (for systems that are purchased and not deployed using the OOB model).

Software distribution. Adding applications and software outside of the imaging cycle, for example, to fulfill specific departmental requests for applications.

Data management. Providing data mobility for end users, backup solutions, and data storage for collaboration services.

Usage management. Shaping the end-user experience in a many:1 (shared use) environment, or using profiles for minimal control in a 1:1 deployment. Also known as client management.

License management. Keeping track of site-licensed software, Volume Purchase Program (VPP) codes, and redemption codes provided to end users.

Patches and upgrades. Keeping Mac systems up to date with the most current security patches, updating the OS and applications with needed improvements, and pushing new drivers for printers.

Help desk management. A formal process and mechanism to provide end-user assistance, training, and support. Also a valuable source of feedback for planning.

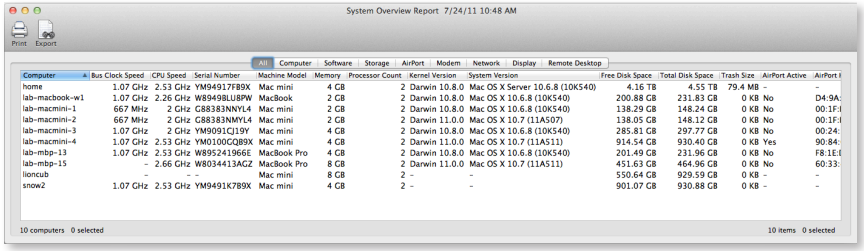
The following section presents best practices and available resources associated with each of these tasks.

Asset management

Third-party systems management solutions can also provide excellent asset-tracking capability:

- Absolute Manage from Absolute Software
- FileWave Management Suite from FileWave
- Casper Suite from JAMF Software
- K2 suite from Sassafras Software

What's on your Mac systems and where is it located? Which systems can be upgraded to OS X v10.8 Mountain Lion? Keeping track of all the hardware and software your organization has purchased or acquired is the responsibility of IT. To plan upgrades, you also need to know, at a glance, which systems have enough RAM for the new installations.



Apple Remote Desktop (ARD) provides tools for tracking your Mac systems. Using an ARD Task Server, you can store system information for hundreds of Mac computers in a central database.

Imaging

Third parties that provide complete imaging solutions include the following:

- Absolute Software
- FileWave
- JAMF Software (Casper Imaging Suite, a subset of the full Casper Suite)

Creating, maintaining, and deploying images is an important part of systems management. The required number of images depends on many factors. Are you imaging portables or desktops? Servers or clients? Basic-use systems or high-end workstations? A successful imaging process starts with proper configuration of images and extends to a procedure for deploying them—whether manually or automatically, in person or across a network.

Details on imaging can be found in the "Image and Deploy" section, which begins on page 20.

Software distribution

Key third-party providers also have solutions for ad hoc software distribution, as well as for self-service distribution:

- Absolute Software
- FileWave
- JAMF Software

Once you've applied core system images, you may need to customize them with special applications and settings for different sites.

Apple Remote Desktop provides tools for deploying additional software post-imaging using one of two methods. The first is to select a group of Mac computers and send them a complete application. If the application is self-contained, (as with the applications purchased from the Mac App Store), the sysadmin can download an institutional copy of the application, and distribute it to all Mac computers covered by the VPP license.

The second method relies on PackageMaker (see page 23) to create an installer for an application bundle. If the application is already in .pkg format, the sysadmin can use Apple Remote Desktop to send the package to the chosen Mac systems. To expedite the process, the sysadmin can send it to an ARD Task Server, which delivers the package to the selected Mac systems as they come online.

Learn more about how to use Apple Remote Desktop for software distribution.

In 1:1 deployments, you may prefer to install software outside of the institutional Mac image. One method of self-service distribution is to provide end users with application redemption codes purchased by the institution. In this scenario, the end user must be the local administrator of their Mac to redeem the code and install the application.

Another self-service option is to package the application and post it on a file or web server for download. If the package is configured properly, a nonadministrative user can install the application in the Applications folder within their own home directory—instead of in the main Applications folder, which would require admin access.

Data management

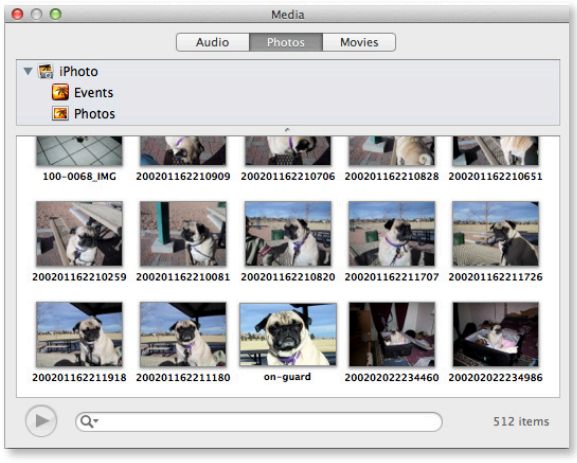
Responsibility for maintaining files needed for day-to-day operations may rest with the end user, institutional IT, or both. Data management processes may be as simple as asking users to archive key files onto a flash drive, or as complex as providing a centralized backup solution for a 1:1 Mac deployment for thousands of users. This section covers two types of data management: data mobility and data backup.

Data mobility

In a deployment where the institution owns the Mac systems and the users are responsible only for their own files, data mobility provides users with easy, anytime access to their work. For example, when deploying a cart of MacBook Air notebooks, a user may be required to log in to a Mac system using a guest account, which means all the files created in that local home directory are deleted on logout. To retain any work during a session, users must copy files to an external location—whether a network file share, an external device such as a USB flash drive, or Internet cloud storage.

When the user account is defined as an external mobile account (see "Assigned Systems" on page 18), data mobility is automatic. The user's home directory resides on the external storage device; as long as the user logs in to a properly managed, bound Mac, their home directory will always be available.

Institution curriculum and technology leaders must also recognize the importance of data mobility when it comes to individualized rich media. Users need to access their creative media files: an iPhoto library, an iTunes library, and any other rich media for creating content, such as video and sound files or artwork. The Media Browser in Pages and Keynote facilitates access to both iTunes and iPhoto libraries, enabling the user to add individualized components to documents and presentations. However, this easy access requires a 1:1 deployment with the user as the local admin or a many:1 deployment with the mobile account option.



Media Browser in Keynote and Pages

Data backup

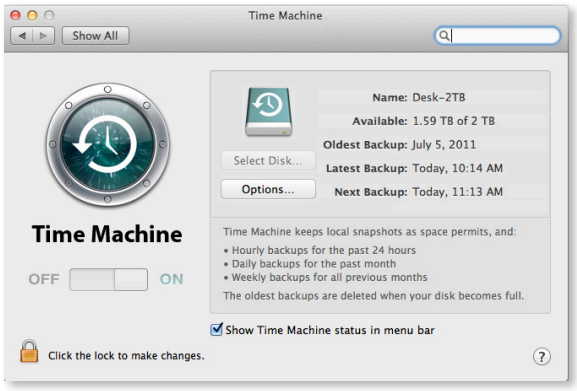
Most educational institutions rely on two different backup workflows. The first is for backing up the school's servers to preserve the data stored on them. The second is for backing up all end-user data, wherever it might be located. In the past, these workflows were the same, since user data was usually found only on school servers. In this legacy scenario, users copied important files onto floppy disks, later to USB flash drives, and more recently, if permitted, to an Internet cloud storage service.

With the growth of 1:1 deployments, it's important to clarify a separate plan for backing up user data. While available solutions cover a wide range of possibilities, following are a few of the best practices.

Self-backup

It may seem a ridiculous idea simply to tell users to back up their own data, but this may be your only option in a 1:1 deployment of thousands of Mac computers. Consider the following scenario, for example: A 1000-unit MacBook Air deployment with the baseline 64GB SSD drive allows each user approximately 35GB of personal space after a full install of iWork, iLife, and some other key applications. To back up that much user space online, you would need 35TB (terabytes) of network storage. If you choose to place a quota on a user's backup space, you will likely be unable to include their iPhoto and iTunes libraries—rendering your backup capabilities of limited value to your users.

An end user who is also local admin can set up their own Time Machine backup with a midsize USB external hard drive for about $50. This would allow the user to maintain a complete backup of their entire home directory, while also reinforcing personal responsibility for their data.



Third-party solutions

Several third parties—including CrashPlan PRO from Code42—offer well-designed backup solutions for both servers and end users. Consider choosing a solution that provides both remote and local backup for end-user data, enabling users to retrieve important files from an Internet-accessible location, from a direct-connect device, or both.

Time Machine server

Smaller 1:1 deployments—limited to a few dozen, or perhaps as many as a hundred users—warrant the use of a Time Machine server. A current Mac mini server with a Pegasus RAID can provide network-based Time Machine backup capabilities for about 100 end users. This solution will not support nonadmin users in a many:1 deployment, however, as all users must be local administrators.

Usage management

How much management is required to meet the educational needs of your institution? Usage management, also referred to as client management, involves establishing an appropriate, consistent end-user experience on a Mac.

In a many:1 deployment, IT staff may choose to lock down a Mac so that the user experience in first period on Monday is maintained all the way through seventh period on Friday—without requiring technical assistance.

For an opt-in experience in a 1:1 deployment, you may simply download a profile to a user's Mac so the user can then connect to the school's Wi-Fi network.

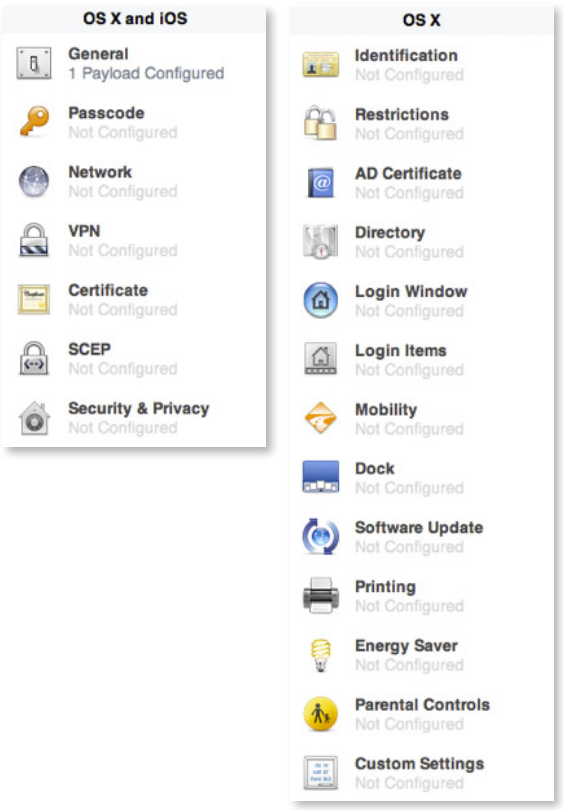
Following are some usage management tools to consider:

- Profile Manager, for deployments where all Mac systems are running OS X Lion or later.
- Workgroup Manager, for environments with a mix of Mountain Lion and Lion-based systems and older Mac computers running an earlier version of OS X (10.6 and earlier).
- A combination of the above, if your deployment includes iOS devices, OS X Mountain Lion–based systems, and older Mac computers: Profile Manager for Mountain Lion and Lion-based Mac systems and iOS devices, and Workgroup Manager for older Mac systems.

You'll want to test your choice of management tools before deployment.

Profile Manager

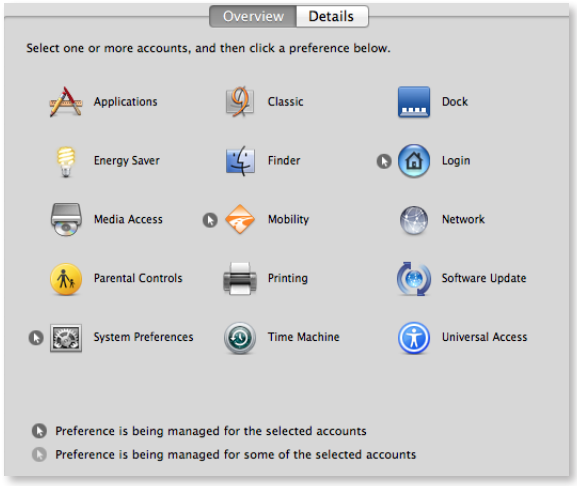
With OS X v10.7 Lion, Apple introduced the ability to manage Mac systems using profiles instead of through directory services. Profiles can contain the same information as the managed client settings in Workgroup Manager. However, profiles don't need to be bound to any directory services and can be provided as opt-in services, where the user chooses to have the profile(s) applied.



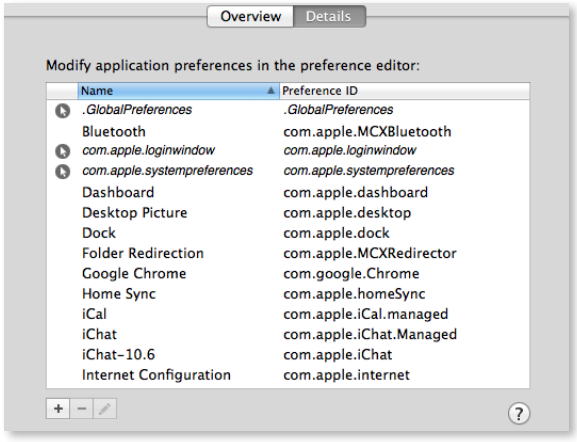
Workgroup Manager

Workgroup Manager, a component of the OS X Server Admin Tools under Lion and a separate download under Mountain Lion, is the best tool for managing a mixed environment of Mountain Lion-based systems and older OS X systems. Using Workgroup Manager in conjunction with OS X Server, you can then set Mac systems to follow a series of preference settings stored in Open Directory. The Mac systems are bound to the Open Directory server and pick up the managed settings either at the login window or when the user logs in, depending on the settings applied. Managed client settings can be applied to computers, groups of computers, users, and groups of users, depending on need.

The Workgroup Manager interface presents a core set of preferences and settings, as shown below.



Click Details to select additional managed client settings and to import settings and preferences from external sources.



In an environment where Active Directory provides user account identification and authentication, you can manage preferences by joining the Mac systems to both Active Directory and Open Directory servers.

Third-party solutions
The following third-party solutions can also help with Mac management:

- Absolute Manage from Absolute Software
- FileWave
- Casper Suite from JAMF Software

Differences between Profile Manager and Workgroup Manager

The key difference between Profile Manager and Workgroup Manager is the change from directory-based management to profile management. Profile Manager allows for profiles that are downloaded or installed on a client Mac. The Mac can then be removed from direct network management, reducing the communication need between the client and the network servers. In a 1:1 deployment, this capability can greatly enhance the user's workflow experience.

Another difference is the total number of management settings available from each tool. Although Profile Manager is technically capable of providing almost every setting that can be created in Workgroup Manager, the settings must be provided in a property list (plist) format. In Workgroup Manager, the Details pane features a predefined set of management options, some of which did not migrate to Profile Manager.

As a result, system administrators will need to evaluate the settings required for their deployment to determine the best management options. It is also important to note that many of the older settings used in Workgroup Manager may no longer apply to a more modern deployment workflow scenario.

Many of the client management settings that existed in Workgroup Manager/Open Directory-based management have the same domains and keys. In directory-based management, settings would be pulled from user/group/computer records and composited on the client. With Profile Manager-based management, these settings are put into a configuration profile and installed on the client either manually (by double-clicking the .mobileconfig file) or pushed from a MDM (for example, Profile Manager) server, if the client has previously installed an enrollment profile. A client may be enrolled either manually (by visiting the Profile Manager website), by installing an enrollment profile in a special location on the client, or by making an enrollment profile part of the imaging process. The latter procedure is intended as an automated way to bind client computers in an educational environment.

Similar to WGM, Profile Manager has dedicated UI for some settings (such as Dock, Media Access, Login Window). It also provides support for setting arbitrary preferences, by way of the Custom Settings payload.

Profile Manager stores the settings for devices, device groups, users, groups, and more on the server itself, in a private database. Unlike OD-based management, the managed client software never pulls the settings from the server's databases directly. Instead, profiles must either be manually exported from Profile Manager or pushed from the server to the client as a result of binding the client to the Profile Manager server.

On the client, the profiles are stored in var/db/ConfigurationProfiles, but this data is not meant to be accessed directly by anyone other than the managed client software. Apple does provide a profiles command line tool for querying/installing/removing profiles.

Additional information can be obtained through the Profiles and Managed Client System Profiler reports. The Profiles tab shows information about installed configuration profiles. The Managed Client tab shows information about composited managed preferences (much as it has in previous releases).

With OD-based management, the vast majority of preferences were composited to a managed preference domain or key that lived in /Library/Managed Preferences and then various components in the system queried those preferences. With configuration profiles, many settings (especially the new ones) are configured directly at the time the profile is installed and may not even have corresponding settings in Managed Preferences.

For example, when old-style settings (such as Dock, Media Access, Login Window) are delivered through a configuration profile, corresponding Managed Preferences are still created on the client, just as they were under OD-based management. The Dock, SystemUIServer, and Login Window still make calls to CFPreferences and pick up the values, just as they have always done.

New settings (such as Mail, iChat, System Policy) configure the options directly. For example, installing a profile with a Mail payload makes calls to the InternetAccounts framework to set up the account immediately. Installing a profile with a CardDAV account makes calls into the AddressBook framework to set up the account. Installing a profile with a System Policy payload modifies the system database directly. In these cases, no corresponding data will exist in the Managed Preferences directory.

For more information, view the following videos on the Apple website:

- Introduction to Profile Manager
- Managing Macs with Lion Server

License management

Third-party solutions for license management range from reporting on applications installed to controlling how many copies of an application can be used at any one time. Following are some of those solutions:

- Absolute Manage from Absolute Software
- Inventory from FileWave
- Casper Suite from JAMF Software
- K2 suite from Sassafras Software

It is IT's responsibility to track, manage, and control the institution's software licenses. Although license management provides operational benefits, it is a legal necessity to make sure no one has deployed more copies of a specific application than you have purchased.

The Mac App Store makes it easy for schools and institutions to manage and maintain application licenses. Learn more about Apple Software Volume Licensing.

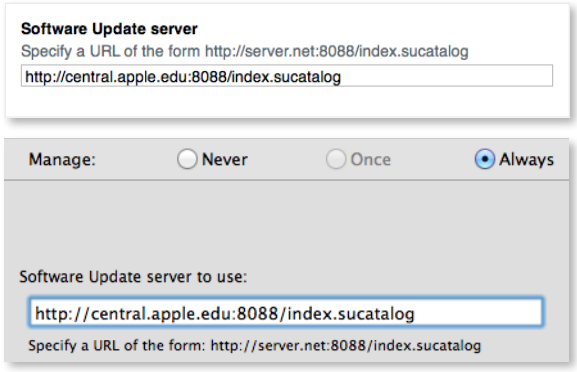
The Apple Remote Desktop report format can locate and track applications installed on Mac systems, as well as report on application usage.

Patches and upgrades

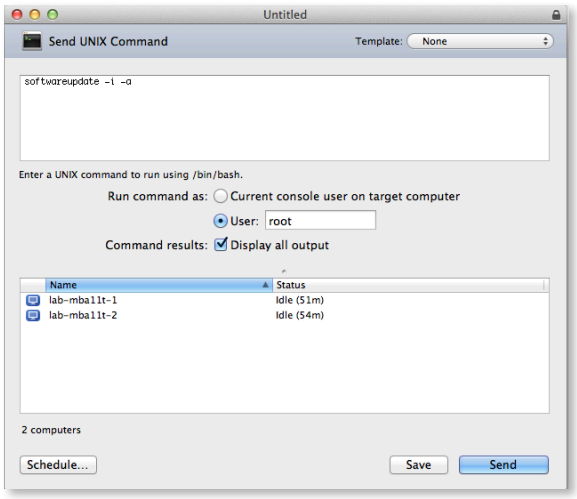
Unlike imaging and software deployment, patching and upgrading tasks take place on an ongoing basis and seldom require a complete rebuild of a client Mac.

Software Update Server and the Mac App Store are the primary mechanisms for updating Apple software and applications. Current OS updates, driver patches, and some key applications, such as Safari, are updated through the software update mechanism. However, in a school with dozens or hundreds of Mac computers, individual software updates can place a tremendous burden on the network. In these environments, consider using OS X Server running the Software Update Server (SUS). This server-side process collects any updates that Apple publishes and provides them inside your institution's firewall.

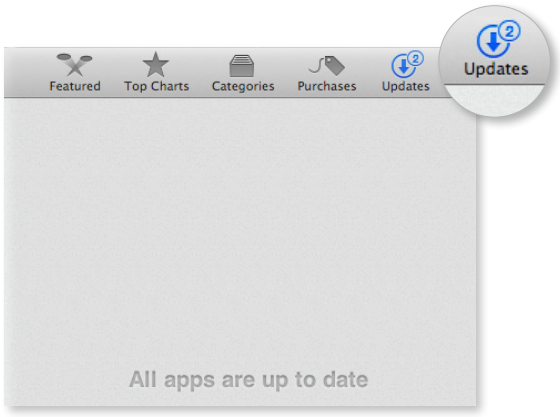
You can use managed client settings in Profile Manager or Workgroup Manager to point your Mac clients to the local SUS, as shown below.



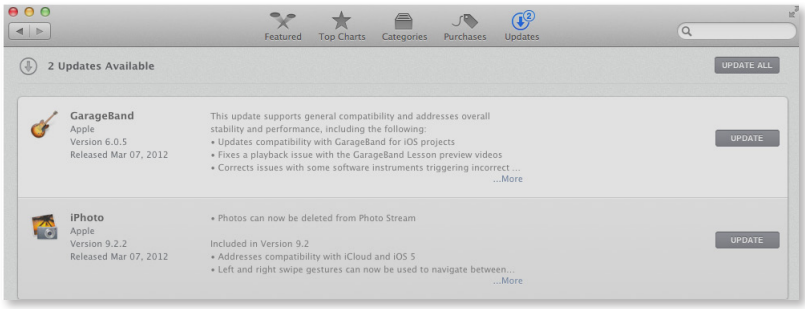
If users are local administrators of their Mac computers, they can run software updates from their own systems. If the computers are shared, IT can mass-deploy software updates using a UNIX command built into Apple Remote Desktop, instructing client systems to check for updates and to install them in the background. In this case, no user interaction is required.



For applications from the Mac App Store, the process for patching and updating is different. Only local administrators can download application updates from the Mac App Store. In the case of a large institutional deployment, it isn't feasible for IT staff to manually log in to each Mac and download the latest updates.



Users will see that updates are available for applications installed using the institution's Apple ID. However, they will not be able to see which applications need updating unless the local administrator logs in to the Mac and opens the Mac App Store with the institutional Apple ID.



Many tools mentioned in this document can assist the help desk, including the following:

- Apple Remote Desktop for remote assistance, installation issues, and device tracking
- Absolute Manage, Casper, and FileWave for deployment assistance and remote assistance

Web Help Desk is a formal help desk solution that provides a scalable, full-service help desk operation. It includes web-based trouble tickets, issue tracking, technician scheduling, and user surveys.

In this scenario, consider having IT use a single master Mac and log in to the Mac App Store with the institution's Apple ID. Next, access the Updates pane and download any updates posted. Locate the updated applications in the Applications folder, and follow the process for deploying Mac App Store applications outlined in the "Applications and packages" section (see page 22)—creating one or more PackageMaker sets for the new versions of the applications.

Once you have created the application packages, you can deploy them to the institutionally controlled Mac systems using Apple Remote Desktop or any third-party package deployment mechanism.

This process of aggregating updates also works with normal Apple Software Updates. You can download all current updates posted at support.apple.com/downloads and collate them into a meta-package using PackageMaker—essentially, adding multiple package installers into a single package for deployment. Learn more about this workflow in PackageMaker User Guide, found in the Mac Developer Library.

Help desk management

A formal help desk establishes a central location for all assistance requests, a clearing house for technical issues, and a formal end-user training program. Help desk services can include end-user mentoring, equipment repair, software installation assistance, and technical staff management.

Conclusion

When planning, evaluating, and architecting a Mac deployment, consider completing the following several evaluations:

- Properly defining the education mission of your institution.
- Examining how the current infrastructure fits that mission.
- Identifying improvements needed to bring the infrastructure up to speed.
- Establishing user account models based on perceived ownership and workflows.
- Determining what systems management solutions to introduce or to enhance.

All of these considerations will impact the overall effectiveness of your deployment. Your Mac deployment can be even more successful when you rely on the expertise of the Apple education sales team, Apple Professional Services, and Apple Professional Development groups.

References

Apple Pro Training series

For more information on disk images and the imaging process, see Apple Pro Training series books: *OS X Lion Support Essentials* and *OS X Lion Server Essentials* by Peachpit Press.

For detailed information on automating common deployment tasks when imaging Mac clients, see the Apple Pro Training series book: *Mac OS X Deployment v10.6* by Peachpit Press. While this book was written for Snow Leopard, it is still valid for Lion deployments.

These books are all available for purchase and download in the iBookstore.

Additional resources

OS X Mountain Lion: About Thunderbolt

OS X Server: NetBoot clients cannot start up from the server (NetBoot troubleshooting)

About Thunderbolt to Thunderbolt cable (2.0 m)

MacBook Air EFI Firmware Update 2.2

MacBook Air EFI Firmware Update 2.4

Mountain Lion Server: Advanced Administration

Disk Utility Help for OS X Mountain Lion

man asr (also viewable from Terminal by typing `man asr` at the prompt)

Appendix A: Wi-Fi Standards

This section discusses the Wi-Fi standards related to designing a Wi-Fi network that will include iOS devices. The selection of each Wi-Fi standard affects the user experience, so a summary of the standards is included.

2.4GHz vs. 5GHz

Wi-Fi networks operating at 2.4GHz allow for 11 channels in the United States. However, due to channel interference considerations, only channels 1, 6, and 11 should be used in a network design.

5GHz signals do not penetrate walls and other barriers as effectively as 2.4GHz signals do, which results in a smaller coverage area. Therefore, 5GHz networks may be preferred when designing for a high density of devices in an enclosed space, such as in classrooms. The number of channels available in the 5GHz band varies among access-point vendors and from country to country, but at least eight channels will always be available.

5GHz channels are nonoverlapping, which is a significant departure from the three nonoverlapping channels available in the 2.4GHz band. When designing a Wi-Fi network for a high density of iOS devices, the additional channels provided at 5GHz become a strategic planning consideration.

IEEE 802.11b/g

If devices that support only the 802.11b or 802.11g standard are required to participate on the network, 802.11b/g should be included in the Wi-Fi network design.

802.11b provides data transfer speeds of up to 11 Mbps, while 802.11g provides data transfer speeds of up to 54 Mbps. Under ideal conditions, the actual data throughput, or the actual speed at which devices will exchange information, is about half the data rate. Both technologies are implemented in the 2.4GHz band, the same band at which many cordless phones, microwaves, and other wireless devices operate. Note that when 802.11b devices and 802.11g devices are using the same wireless network, the 802.11b devices cause reduced data throughput for the faster 802.11g clients.

IEEE 802.11a

In contrast to 802.11b/g, the 802.11a standard operates in the 5GHz band. Most notebooks support this band, but many smaller mobile devices support 2.4GHz Wi-Fi only. Transfer speeds and data throughput when using 802.11a are similar to those with 802.11g.

IEEE 802.11n

The newest 802.11 standard is 802.11n, which is capable of transmit speeds of up to 600 Mbps. To accomplish this task, several technologies are used.

802.11n can use either the 2.4GHz or 5GHz band and is compatible with 802.11a/b/g standards, so older devices can share the same network with newer 802.11n devices.

802.11n supports several operating modes:

- 802.11n at 5GHz
- 802.11n at 2.4GHz
- 802.11n + 802.11a at 5GHz
- 802.11n + 802.11b/g at 2.4GHz
- 802.11n + 802.11g at 2.4GHz
- 802.11n + 802.11b at 2.4GHz

Most multiradio access points allow any combination of the above modes.

The 802.11n standard uses a technology called multiple-input multiple-output (MIMO) to achieve higher speeds. MIMO supports the transmission of multiple streams of data, called spatial streams, simultaneously. To take advantage of these spatial streams, both the access point and client must have multiple radios and antennas. Mac products support multiple spatial streams while iOS devices support a single spatial stream.

HD40, commonly referred to as wide channels or channel bonding, is another technology used to accomplish faster transmit speeds. Approximately double the amount of data can be transmitted through this single, but wider, channel. Nonbonded channels are called HD20. Channel bonding should not be used in the 2.4GHz band because only three nonoverlapping channels are available. Thus, many access point vendors do not support configuring channel bonding when using the 2.4GHz band. iOS devices support HD20, while Mac products support channel bonding.

Wi-Fi standards support in Apple products

The chart below shows the Wi-Fi specifications supported by Apple products. The list includes the following details:

- 802.11 compatibility: 802.11b/g, 802.11a, 802.11n.
- Frequency band: 2.4GHz or 5GHz.
- Modulation and Coding Scheme (MCS) index: Defines the actual data rate at which 802.11n devices can communicate. See the MCS index table in this appendix for more information.
- Channel bonding: HD20 or HD40.
- Guard interval (GI): The space (time) between symbols or characters of information transmitted from one device to another. The 802.11n standard defines a short-guard interval of 400ns that allows for faster overall throughput, but devices can use a long-guard interval of 800ns.

iPhone 5
- 802.11n at 2.4 GHz and 5 GHz
- 802.11 a/b/g
- MCS Index 7 / HD40 / 400ns GI

iPhone 4/iPhone 4S
- 802.11n at 2.4GHz
- 802.11b/g
- MCS Index 7/HD20 / 800ns GI

iPhone 3GS
- 802.11b/g
- MCS Index 7/HD20 / 800ns GI

iPad mini, iPad 2, and iPad with Retina display
- 802.11n at 2.4GHz and 5GHz
- 802.11a/b/g
- MCS Index 7/HD20 / 800ns GI

iPod touch (5th generation)
- 802.11n at 2.4GHz and 5 GHz
- 802.11a/b/g
- MCS Index 7/HD20 / 800ns GI

iPod touch (4th generation)
- 802.11n at 2.4GHz and 5 GHz
- 802.11a/b/g
- MCS Index 7/HD20 / 800ns GI

MacBook Pro, MacBook Air, and MacBook Pro with Retina display
- 802.11n at 2.4GHz and 5GHz
- 802.11a/b/g
- MCS Index 15 / HD40/400ns GI
- MCS Index 23 / HD40/400ns GI (early 2011 or later MacBook Pro)

MCI index

MCS index	Spatial streams	Modulation	Coding rate	Data rate (in Mbps) (GI = 800ns)		Data rate(in Mbps) (GI = 400ns)	
				20MHz	40MHz	20MHz	40MHz
0	1	BPSK	1/2	6.5	13.5	7.2	15.0
1	1	QPSK	1/2	13.0	27.0	14.4	30.0
2	1	QPSK	3/4	19.5	40.5	21.7	45.0
3	1	16-QAM	1/2	26.0	54.0	28.9	60.0
4	1	16-QAM	3/4	39.0	81.0	43.3	90.0
5	1	64-QAM	2/3	52.0	108.0	57.8	120.0
6	1	64-QAM	3/4	58.5	121.5	65.0	135.0
7	1	64-QAM	5/6	65.0	135.0	72.2	150.0
8	2	BPSK	1/2	13.0	27.0	14.4	30.0
9	2	QPSK	1/2	26.0	54.0	28.9	60.0
10	2	QPSK	3/4	39.0	81.0	43.3	90.0
11	2	16-QAM	1/2	52.0	108.0	57.8	120.0
12	2	16-QAM	3/4	78.0	162.0	86.7	180.0
13	2	64-QAM	2/3	104.0	216.0	115.6	240.0
14	2	64-QAM	3/4	117.0	243.0	130.3	270.0
15	2	64-QAM	5/6	130.0	270.0	144.4	300.0
16	3	BPSK	1/2	19.5	40.5	21.7	45.0
17	3	QPSK	1/2	39.0	81.0	43.3	90.0
18	3	QPSK	3/4	58.5	121.5	65.0	135.0
19	3	16-QAM	1/2	78.0	162.0	86.7	180.0
20	3	16-QAM	3/4	117.0	243.0	130.0	270.0
21	3	64-QAM	2/3	156.0	324.0	173.3	360.0
22	3	64-QAM	3/4	175.5	364.5	195.0	405.0
23	3	64-QAM	5/6	195.0	405.0	216.7	450.0
24	4	BPSK	1/2	26.0	54.0	28.9	60.0
25	4	QPSK	1/2	52.0	108.0	57.8	120.0
26	4	QPSK	3/4	78.0	162.0	86.7	180.0
27	4	16-QAM	1/2	104.0	216.0	115.6	240.0
28	4	16-QAM	3/4	156.0	324.0	173.3	360.0
29	4	64-QAM	2/3	208.0	432.0	231.1	480.0
30	4	64-QAM	3/4	234.0	486.0	260.0	540.0
31	4	64-QAM	5/6	260.0	540.0	288.9	600.0

Appendix B: Wireless Security

Over time, several technologies have been developed to protect and secure Wi-Fi networks. Some of the early technologies include WEP (Wired Equivalent Privacy), LEAP (Lightweight Extensible Authentication Protocol), device filtering by MAC address, and hiding the network SSID. Although these technologies did provide some level of Wi-Fi network security at the time, they are no longer considered secure and can be easily compromised.

Fortunately, current Wi-Fi standards such as WPA and WPA2 provide technologies for network authentication and encryption to secure data. If these security standards are in place, there is no benefit in implementing legacy technologies.

IEEE 802.11i, WPA, and WPA2

WPA (Wi-Fi Protected Access) and WPA2 refer to a suite of tests that ensure compatibility between various Wi-Fi devices. The actual Wi-Fi security standard is defined by the IEEE in 802.11i. In broad terms, this specification defines two areas of network security: authentication for obtaining access to the network and encryption of data itself as it passes from one Wi-Fi device to another. WPA and WPA2 are commonly used to define which 802.11i options are enabled on the network. The main difference between WPA and WPA2 is the strength of data encryption. WPA2 is preferred over WPA.

PSK vs. enterprise

Access to a WPA or WPA2 network can be secured with a single password for all users, or by providing an individual credential to each user or device. This credential could be either a user name and password, or a PKI identity (certificate). Using a single password for all devices is referred to as a Pre-Shared Key (PSK). The enterprise version refers to the implementation of 802.1x for individual credentials assigned to each user or device. Regardless of the method used for network authentication and encryption, be sure to use WPA or WPA2 for a secure Wi-Fi network.

Broadcast or hidden SSID

A Wi-Fi network name is called the Service Set ID, or SSID. To join a specific wireless network, the user selects the SSID for the desired network from a list of SSIDs being broadcast within the range of the Wi-Fi device. However, it's also possible to hide the SSID so that it does not show up in searches. While there may be a perception that hiding the SSID is more secure than broadcasting the SSID, there is very little security benefit.

Hiding the network SSID means that a user won't see the network in a list of networks within range of the computer, but it would take a potential hacker only a few seconds to obtain the name of the network simply by using a computer to listen to information being transmitted by Wi-Fi devices already associated with the hidden SSID. This is possible because even with a hidden SSID, the name of the network is transmitted unencrypted within the data.

More important are the practical implications of a hidden SSID. For a Wi-Fi device to rejoin a hidden SSID, it must first locate access points offering that SSID. However, because the SSID is hidden, the Wi-Fi device must visit every known channel and broadcast to see if the hidden SSID exists on that channel. After broadcasting, the

computer must wait a certain amount of time for responses. If the client has multiple saved hidden SSIDs, it must broadcast on each channel for each of the SSIDs and wait for a response after every channel broadcast for every SSID.

When finding a broadcasted SSID, the computer visits each channel and simply listens for the SSIDs that exist on that channel. It doesn't matter how many saved broadcast SSIDs might exist on a computer, the computer still needs to listen only one time on each channel to find them.

Simply put, it's harder for a Wi-Fi device to rejoin a hidden SSID than a broadcasted SSID, and there's very little security benefit in hiding the SSID. iOS devices tend to physically move frequently, so hidden SSIDs may delay their network association time.