Mac OS X for UNIX Users

The power of UNIX. The simplicity of Macintosh.

Features

Open source UNIX® foundation

- POSIX compliant, Open Brand UNIX 03 Registered Product
- Open source kernel based on FreeBSD and Mach 3.0
- 64-bit virtual memory using LP64 data model
- Advanced support for multiple CPU cores
- Hand-tuned, standards-compliant scalar and vector math libraries
- UNIX GUI support via native toolkits and bundled X11 server

Standards-based networking

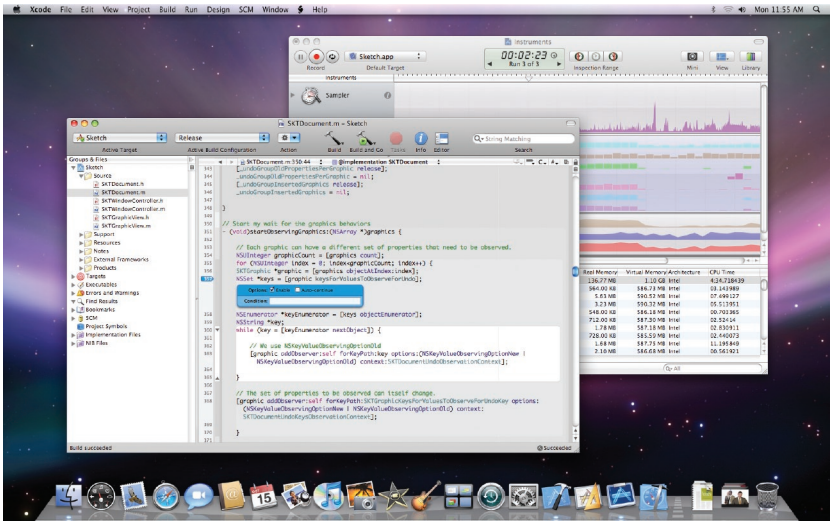
- Complete IP-based architecture supporting IPv4, IPv6, and L2TP/IPSec VPN
- Rich zero-configuration discovery and naming via Bonjour and Dynamic DNS
- Interoperable file serving via NFS, AFP, SMB/CIFS, and FTP
- Powerful Apache services (HTTPD, DAV, PHP)
- Open directory services built on LDAP and Kerberos for single sign-on

Comprehensive UNIX user environment

- Standards-based graphics built on PDF (Quartz), OpenGL, and H.264 (QuickTime)
- Brand-new tabbed, internationalized Terminal
- Capability to develop and deploy Ruby on Rails web applications
- Scripting of Cocoa objects and Apple events from both Ruby and Python
- Comprehensive UNIX/Linux utilities (such as emacs, vim, gnutar, make)
- Free bundled Xcode 3 developer tools with GCC 4 and DTrace-based instruments

Mac OS X version 10.5 Leopard combines a fully conforming UNIX foundation with the richness and usability of the Macintosh interface, bringing multicore technology and 64-bit power to the mass market. The new tabbed Terminal, Ruby on Rails web application stack, and powerful scripting bridges make UNIX users more productive than ever.

There are already tens of millions of Mac OS X users—consumers, scientists, animators, developers, system administrators, and more—making Mac OS X the most widely used UNIX desktop operating system. In addition, Mac OS X is the only UNIX environment that natively runs Microsoft Office, Adobe Photoshop, and thousands of other consumer applications—all side by side with traditional command-line, X11, and Java applications. Mac OS X is also the foundation for Mac OS X Server, which makes open source easy to administer. Tight integration with hardware from the sleek MacBook notebook computer to the eight-core Mac Pro computer is making Mac OS X the platform of choice for an emerging generation of UNIX users.

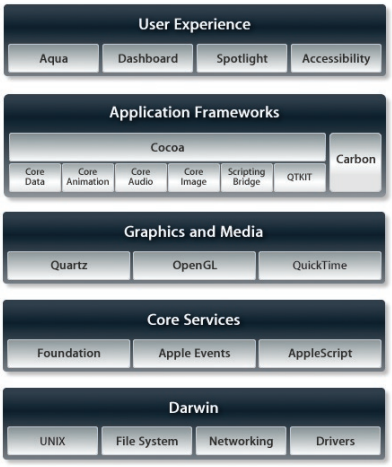


The complete Xcode tools suite includes everything you need to develop for the Mac.

Top New Features for UNIX Users

- **UNIX certification.** Leopard is now an Open Brand UNIX 03 Registered Product, conforming to the SUSv3 and POSIX 1003.1 specifications for the C API, Shell Utilities, and Threads.
- **64-bit applications.** Run next-generation applications (in OpenGL, X11, or Cocoa) that can address massive amounts of memory, on the exact same operating system and drivers you use today.
- **Multicore optimized.** Take full advantage of newer Mac computers with multiple CPU cores, thanks to smarter scheduling and thread management.
- **DTrace.** Developers and system administrators can rapidly profile C, Java, Perl, Python, and Ruby processes, thanks to integration of the DTrace Open Source project into the Mac OS X kernel.
- **Ruby and Python development.** Use Ruby and Python as first-class languages for building applications with Cocoa or automating applications using Apple events. Leopard also enhances the status of Mac OS X as the premier platform for Ruby on Rails development by bundling Rails, Mongrel, and Capistrano.
- **Streaming I/O.** Developers can easily set up high-bandwidth data transfers (for example, DMA or pro video) without worrying about the underlying memory architecture.
- **Self-tuning TCP.** The system automatically adjusts the TCP buffer size for optimum performance, which provides significant benefits in high-bandwidth/high-latency networks.
- **Terminal 2.** The brand-new Terminal.app now takes full advantage of Mac OS X native text and graphics capabilities, using Input Manager and CoreText to fully support non-English languages. A modern tabbed interface makes it easy to manage multiple terminal sessions without window clutter.
- **Autofs.** Mac OS X automatically mounts and dismounts network file systems (AFP, NFS, SMB) on separate threads to improve responsiveness and reliability.
- **Kerberized NFS.** Now you can securely connect NFS clients and servers using Kerberos.
- **Wide-Area Bonjour.** Use a single, consistent host name to locate your Mac, whether behind a NAT gateway or hopping across DHCP servers.
- **Directory Utility.** A single place to graphically manage all local and remote directory entries and services, which otherwise would require complicated command-line operations.

Mac OS X Architecture



Modular architecture of Mac OS X

Mac OS X flexibility is derived from a modular architecture built around five major layers:

User experience
The entire interface—including icons, menus, windows, and controls—represents an innovative continuation of the legendary Mac ease of use, using color, transparency, and animation to enhance the usability and consistency of the system and applications. As part of that, Mac OS X bundles over 350 high-quality graphical applications for file management, Internet access, system configuration, and much more.

Frameworks
Mac OS X includes a variety of rich application frameworks, built on top of the traditional UNIX APIs, to support developers in many different communities.

- Cocoa is a set of object-oriented frameworks designed for rapid application development, making it easy to add rich Aqua interfaces to existing UNIX software or to create entirely new applications.
- Carbon provides a gentle migration path for developers using C++ and procedural application frameworks.
- Java 2 Standard Edition on Mac OS X is fully compliant, highly optimized, and tightly integrated with the native look and feel, making it easy to run standards-based Java applications right out of the box.

Graphics
The Mac OS X graphics system combines 2D, 3D, and time-based media standards using an industry-leading compositing window system for a rich, seamless user experience.

- Quartz is the Mac OS X high-performance imaging model, based on Adobe's cross-platform Portable Document Format (PDF) standard. Quartz leverages the GPU to efficiently render high-quality, anti-aliased text and graphics.
- OpenGL is the industry standard for visualizing 3D shapes and textures. Mac OS X features a tightly integrated, highly optimized, and standards-compliant implementation that uses high-end 3D graphics cards to full advantage.
- QuickTime, Apple's cutting-edge digital media software, provides a fully standards-based environment for creating, playing, and delivering video (MPEG-4, H.264), audio (Advanced Audio Coding), and images (JPEG, PNG, TIFF, and hundreds more).

System services
Beneath the easy-to-use interface and rich graphics are powerful system services for directories, mobility, searching, and security. Together, these help ensure that Mac OS X functions consistently, compatibly, and safely wherever you go.

Darwin foundation
Powering all these capabilities is Darwin, an open source UNIX foundation built on technologies such as FreeBSD, Mach, Apache, and GCC. Darwin provides a complete UNIX environment, with X11 and POSIX services comparable to Linux or FreeBSD, including the familiar kernel, libraries, network services, and command-line environment described on the following pages.

Mac OS X: The desktop for Open Source
Since Apple first released the Darwin code in 1999, Mac OS X has been closely identified with the open source community. Apple worked with other pioneers to develop one of the first corporate open source licenses and has built ongoing relationships with key projects such as Apache, Python, and Ruby on Rails. Today, Apple is the largest single vendor of open source software such as GCC, PHP, Samba, SSH, and scores of other projects. To learn more, visit www.apple.com/opensource.

State-of-the-Art Foundation

The Mac OS X kernel at the heart of Darwin is based on FreeBSD 5 and Mach 3.0. Apple has extended this time-tested Mach/BSD foundation with a number of powerful new features, including:

Process and memory management

- **Loadable kernel modules.** Mac OS X dynamically loads appropriate kernel extensions, also known as "kexts," for file system and platform support. This leads to a slimmer kernel that users never need to recompile.
- **Kernel programming interfaces.** Rather than giving kexts direct access to kernel data structures, each version of the kernel exports a well-defined programmatic interface known as a KPI, ensuring third-party kernel extensions can still work even if the internal implementation changes.
- **Efficient kernel threads.** Each POSIX thread is queued onto a particular CPU, improving processor affinity and scalability while reducing lock contention. Threads conform to POSIX (1c), including support for cancellation and shared mutexes.
- **User-level real-time support.** Unprivileged threads can request a fixed number of high-priority cycles per millisecond, for example, for burning DVDs. The kernel ensures that such threads keep their contract.
- **Asynchronous I/O.** Both BSD select and SysV poll are implemented on top of the flexible and scalable kqueue/kevent APIs from FreeBSD 5, handling file system notifications, signals, and much more. File descriptors can also be marked lower priority, for example, for background logging.
- **Launched daemon management.** Mac OS X has replaced init, cron, xinetd, and /etc/rc with launchd, an innovative service that centralizes and simplifies the configuration, management, and monitoring of system, user, and network services.
- **DTrace.** Apple has integrated the DTrace open source project and D language from Solaris into Mac OS X, enabling highly detailed, low-level reports of system and application behavior in languages including C, C++, Objective-C, Java, Perl, Python, and Ruby.
- **Unified memory mapping.** Mac OS X maps both virtual memory and files using the same buffer cache, reducing the total amount of wired memory while improving performance.
- **Virtual video RAM (VRAM).** The GPU's VRAM can be backed by main memory, allowing simpler data transfer and the use of larger data sets.
- **POSIX spawn.** Mac OS X natively implements the posix_spawn system call, which reduces memory requests and improves performance compared with vfork + exec.

System data type sizes (in bytes)

Type	32-bit size	64-bit size
`char`	1	1
`short`	2	2
`int`	4	4
`long`	4	8
`long long`	8**	8**
`float`	4	4
`double`	8	8
`long double`	16	16
`void *`	4	8
`void (*)(void)`	4	8
`size_t`	4	8
`off_t`	8	8

** Native long long operations can be enabled (from 32-bit or 64-bit mode) as described in the "Integrated Developer Tools" section.

64-bit services

Leopard features an upgraded kernel tuned specifically for 64-bit computing, while still natively running 32-bit applications and devices. Key features include:

- **64-bit memory addressing.** Leopard processes have the option of using 64-bit pointers, which on modern chips enable access to many terabytes of virtual (and physical) memory. This is particularly useful for scientific computing and multimedia solutions that need to access huge (greater than 4 GB) data sets.
- **Full 32-bit compatibility.** The Mac OS X kernel is designed to natively run 32-bit and 64-bit applications, maximizing compatibility and performance.
- **Full driver compatibility.** The 64-bit applications can still use existing 32-bit drivers. Both 64- and 32-bit applications can take advantage of drivers upgraded for native 64-bit data transfers.
- **Universal binaries.** A single binary can contain 32-bit and 64-bit binaries for both Intel and PowerPC architectures, so an application shared over NFS would run as 32-bit on a PowerBook G4 but 64-bit on a Mac Pro.

64-bit system libraries

The following APIs are available for use by 64-bit applications written in C, C++, and Objective-C:

- Cocoa (including CoreData, WebKit, and so on)
- Quartz
- OpenGL
- QTKit
- X11
- POSIX (System.framework)
- Libperl, libpython, libruby

There is also a 64-bit Java runtime available for Intel's x86_64 architecture.

- **64-bit inodes.** Files now have 64-bit inode numbers, enabling enormous numbers of files per volume as well as other next-generation file system functionality.
- **64-bit network APIs.** All networking APIs are now 64-bit aware and are used by the 64-bit MySQL database and Apache 2.0 web server (including mod_php) bundled with Mac OS X Server.
- **Optimized math libraries.** The various math and vector libraries have been tuned to take advantage of the latest chip architectures, including support for both 32-bit and 64-bit pointers. See "Numerical Libraries" for more details.

Hardware I/O

I/O Kit is the device driver subsystem of Mac OS X. This powerful, object-oriented architecture in embedded C++ helps device manufacturers rapidly create drivers that run safely in a multiprocessing, preemptive, hot-pluggable environment. Note that every driver on Mac OS X can be used by both 32-bit and 64-bit applications, including drivers that have been upgraded to native 64-bit data transfers.

Key technologies supported by I/O Kit include:

- **Power management.** New power management APIs and notification ordering ensure that devices power down only after their dependent systems have been appropriately notified.
- **Video.** The IOVideo family is designed to support professional video cards and is closely related to the data capture technology used in QuickTime Pro. It is based on the new IOStream class, which provides a high-level API for managing DMAs and other high-bandwidth data transfers without the need to worry about different hardware architectures and optimal caching strategies.

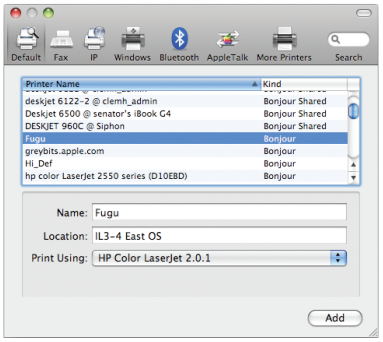
Apple's AirPort brought Wi-Fi to the mass market.

- **USB.** Apple jump-started the market for plug-and-play USB peripherals in 1998 and still leads the industry with built-in support for USB drives, cameras, game inputs, audio, MIDI, and printers. You may never have to download another driver.
- **FireWire.** Apple's award-winning FireWire (IEEE 1394) is a boon to professional audio and video production. Mac OS X provides a comprehensive suite of drivers that take full advantage of the hot pluggability, daisy chaining, and power management capabilities of FireWire.
- **Ethernet.** Since every Mac comes with built-in Ethernet, Mac OS X supports everything from 10BASE-T to 10 gigabits, including automatic link detection, duplex matching, IEEE 802.3ad link aggregation, and even jumbo frames support.
- **AirPort.** Every Mac is designed to work seamlessly with 802.11b/g/a/n networks, such as those provided by the new AirPort Extreme Base Station. Thanks to Wide-Area Bonjour and NAT-PMP support, you can even access your home printer and drives from the public Internet.
- **Bluetooth.** Mac OS X built-in support for a wide range of Bluetooth devices is closely integrated with Leopard Sync Services, so information can be shared seamlessly among devices and one or more Mac computers.
- **Fibre Channel.** Mac OS X provides built-in support for Apple's Fibre Channel cards, the technology of choice for high-performance storage such as Apple's industry-leading Xserve RAID.

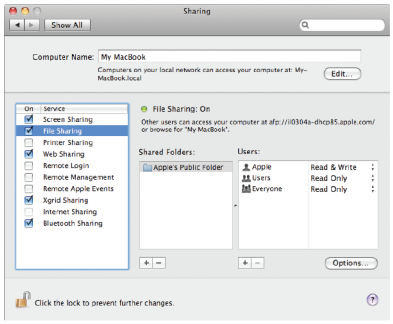
TCP/IP networking

Mac OS X TCP/IP implementation is based on the original BSD networking stack, but over the years, Apple has added numerous powerful enhancements:

- **Self-tuning TCP.** Mac OS X sets the initial max TCP window according to the local resources and connection type, enabling TCP to do a better job of optimizing performance on high-bandwidth/high-latency networks. It also implements Selective Acknowledgment (SACK) for lossy networks, while throttling the bandwidth.
- **Multithreading.** The stack is now fully multithreaded, with at least one input queue per network interface and distinct pools of output message buffers (mbufs) for each CPU, to minimize the need for shared locks.
- **Configuration API.** Rather than manually editing multiple configuration files, developers can use a new high-level C API to programmatically set advanced network preferences.
- **GPRS modems.** A new interface in Network Preferences simplifies the configuration of nondialing (such as GPRS) cell phone modems and will automatically match many users' phones.
- **IPv6.** Apple's low-level networking APIs and high-level applications work seamlessly with IPv6, the next-generation 128-bit Internet Protocol.
- **Internet sharing.** A single click allows your Mac to share its Internet connection over any other interface to one or more computers, using Network Address Translation (NAT) and the Dynamic Host Configuration Protocol (DHCP). This works with a Mac, a PC, and even a virtual system.



Simple interface to manage comprehensive, standards-based printing.



Enable file and Internet sharing with a single click.

- **Apache HTTP.** Mac OS X uses the latest Apache 2.0 for both personal file sharing and high-end web serving, including support for PHP, mod_proxy_balancer, and FastCGI.
- **CUPS printing.** Mac OS X uses a PDF-based printing architecture built entirely around the open source Common UNIX Printing System (CUPS), now owned by Apple. Thanks to a bundled distiller, CUPS can even export local inkjet printers as network PostScript printers for use by Windows systems.

File system architecture

Apple's dynamic implementation of BSD's virtual file system layer (vfs) allows Mac OS X to load numerous local and remote file systems on demand, including:

Disk file systems

- **HFS+,** the default, provides fast Btree-based lookups, robust aliases, and rich metadata (including fine-grained access controls and extended attributes). It also has options for journaled and case-sensitive variants.
- **ISO 9660,** the standard CD-ROM format.
- **UDF 2.5,** the Universal Disk Format, is now fully read/write and works with standard block storage (Flash, hard drive) as well as optical (CD, DVD) devices.
- **FAT32,** the standard Windows interchange format.
- **NTFS,** a new, read-only implementation of the high-end Windows file system format.
- **ZFS** (read-only), a highly scalable file system ported from Solaris.

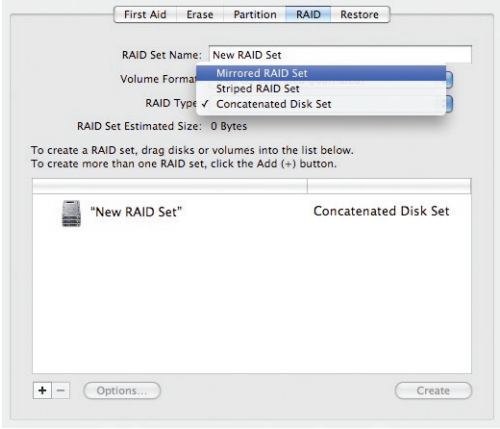
Network file systems

- **AFP,** the Apple File Protocol, remains the principal file-sharing protocol for Macintosh systems due to its robust security and metadata support. It now makes better use of multiple threads and supports remote Spotlight searches.
- **NFS,** the Network File System standard for UNIX, has been dramatically updated with Kerberos support, a multithreaded server, launch-on-demand daemons, use of the nfs.conf configuration file, fine-grained statistics, and numerous performance improvements.
- **SMB/CIFS,** Microsoft's proprietary Server Message Block/Common Internet File System file service for Windows, has been enhanced with Kerberos-enabled automatic reconnects, smbfs packet signing, direct-hosted NetBIOS-less connections, and optional use of NTFS multistream files for Mac resource forks and metadata.
- **FTP** (read-only), the File Transfer Protocol file system.
- **Web Distributed Authoring and Versioning (DAV),** as used for .Mac iDisk.

Control exactly who is allowed to access your files.

File system features

- **Extended attributes.** Leopard extends the xattr routines introduced in Tiger with time-stamped attributes (allowing more efficient backups) and a new high-level copyfile(3) API that automatically transfers all metadata and permissions.
- **Boot Root.** Mac OS X automatically creates a special HFS+J partition containing the last three known good copies of the bootloader, kernel, and essential kexts to enable the use of nonbootable "root" partitions (such as NTFS).
- **Access control lists (ACLs).** The Finder can now set read and write permissions for files and folders on both a per-user and per-group basis. The underlying kauth subsystem can also limit:
  - Read/write/execute/append for either data or extended attributes
  - Read/write (regular) attributes
  - Read/write security settings, or change ownership
  - Add/delete files, directories, or children
  - Search/list directories
- **File locking.** Mac OS X provides unified file locking across AFP, CIFS (SMB), and NFS volumes, simplifying resharing and reducing the risk of data corruption.
- **Network home directories.** Mac OS X supports AFP, SMB, and NFS network home directories, which can all be automatically synced with Mac OS X Server.
- **RAID.** In addition to striping (RAID 0) and mirroring (RAID 1), Apple's built-in software RAID supports combined 0+1 and 1+0 when using four or more disks, providing greater fault tolerance and higher performance.



Easily create redundant arrays of inexpensive disks.

- **Error checking.** The system will explicitly warn of corrupted files (usually by name) and perform automatic "fsck" checks on reboot.
- **fsevents.** The file system events daemon used by Spotlight is now available to user processes, so they can receive immediate notice of modifications.

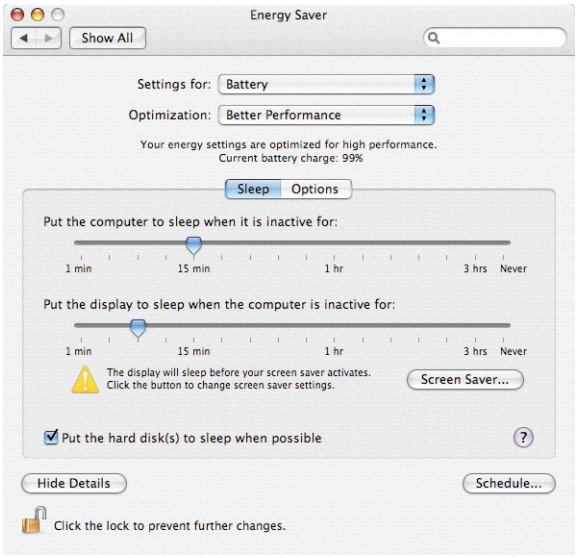
System Services

Mac OS X integrates mobility, directory services, and security throughout the system in order to bring the openness and flexibility of UNIX to the mass market while still ensuring a secure, user-friendly environment.

Mobility

The ability to deal with a constantly changing physical and network environment is one of the key features that distinguishes Mac OS X from other UNIX systems. Beneath the friendly interface for configuring and switching networks lies a very powerful infrastructure.

- **Notifications.** Notify is a comprehensive systemwide service, based on kqueue, for communicating both kernel-to-application and application-to-application events. This reduces the reliance on signals and other legacy UNIX mechanisms used for this purpose.
- **configd.** This unique configuration daemon notifies the system to automatically reroute Internet traffic between wired and wireless connections whenever the user plugs or unplugs the Ethernet cable, as well as to reset other daemons as needed.
- **Power management.** Because Mac OS X is tightly integrated with the underlying hardware, it is the first UNIX system in which instant sleep/wake and power minimization "just work," without the need for manual configuration or determining which chipset is being used.

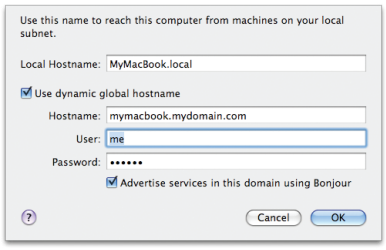


- **Safe disconnect.** Mac OS X is the first system to gracefully mount and unmount NFS, DAV, and SMB/CIFS volumes based on changes to network status or available directory services, providing an uninterrupted experience from either the GUI or the command line.

Directory services

Leopard is designed to minimize configuration effort by pulling information from a variety of sources, including:

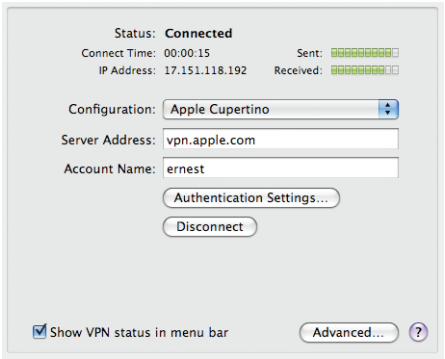
- **Open Directory.** Open Directory is a unified architecture for managing user and system information across the network. While primarily designed for use with LDAP v2/v3 (the IETF standard Lightweight Directory Access Protocol), Open Directory also supports Microsoft's Active Directory and Sun's Network Information System (NIS). Local information is now stored using individual files for better performance and reliability and can be manipulated using passwd and chsh.
- **Bonjour.** Bonjour uses industry-standard zero-configuration networking to help you automatically find printers, file servers, and other network services on your local network. In Leopard, Bonjour has been extended to work through NAT gateways and can be configured to dynamically register a global (for example, the Internet or sitewide) host name using a custom domain, as used by .Mac for Back to My Mac.
- **DNS (Domain Name Services).** Mac OS X uses the Berkeley Internet Name Daemon (BIND 9) to turn DNS host names into IP addresses. The BIND in Leopard Server supports Dynamic DNS and Secure DNS, enabling automatic updates of IP addresses.



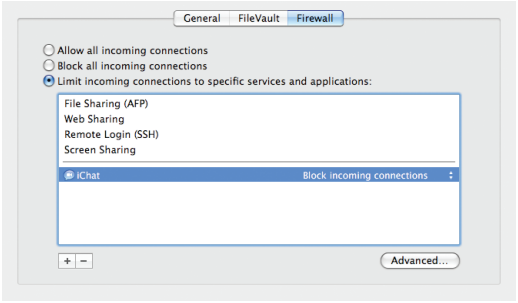
Sharing Preferences let you configure Dynamic DNS for Bonjour over wide area networks.

Secure Networking

- **VPN via L2TP/IPSec or PPTP.** Mac OS X includes a virtual private network (VPN) client that supports the Internet standard Layer 2 Tunneling Protocol (L2TP) over IPSec (the secure version of IPv4), as well as the older Point-to-Point Tunneling Protocol (PPTP). Leopard also adds support for IPSec groups for use with with Cisco VPN servers, as well support for automatic load-balancing, failover, and much better scalability.



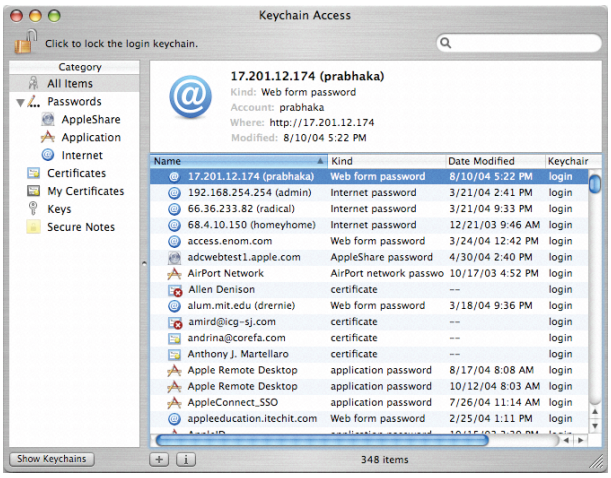
- **SSH (Secure Shell).** Mac OS X now allows OpenSSH to store and retrieve private key pass phrases from a keychain, reducing the need to type a pass phrase. The ssh-agent is now launched on demand for any Aqua login, allowing it to be accessed from any application—for example, any terminal window.
- **Built-in firewall.** In addition to the Ipfw2-based systemwide firewall, Mac OS X includes an application firewall that can be configured to allow only incoming access to preapproved applications and services.



Authentication

Leopard provides a state-of-the art security infrastructure based on modern, standards-based technologies for authenticating who you are and authorizing what you can do. Key technologies include:

- **Systemwide keychains.** On Mac OS X, certificates and user passwords (for example, for mail accounts and websites) are stored in a secure keychain, which provides them only to verified, preauthorized applications. Keychains can be securely synced from machine to machine, and organizations can even specify a site-specific master password to ensure recoverability.



- **Certificates.** Public key infrastructure (PKI) authentication is integrated throughout Mac OS X. PKI keys and other X.509 certificates can be stored in:
  - Smart cards
  - Keychains
  - Address Book
  - LDAP directories

  They can be used for:
  - iChat chats and conferencing
  - IPSec-based VPNs
  - S/MIME email
  - Document signing
  - Numerous other services

  Users purchasing .Mac accounts receive certificates for encrypting iChat messages and enabling single sign-on between registered Mac computers. There's even a lightweight certificate authority in Mac OS X so workgroups can establish their own local web of trust, as well as full certificate management in Mac OS X Server.

- **Unified authentication service.** A single security daemon (securityd) processes authentication requests from both GUI and command-line applications to simplify the implementation, adoption, and auditing of new forms of authentication.

Kerberized services

Client-side only
- loginwindow
- Safari (HTTP, RSS)

Client and server
- Apple File Sharing 1 (AFP)
- DAV* (HTTP)
- Mail** (IMAP/POP/SMTP)
- NFS
- Open Directory** (LDAPv3)
- Samba* (SMB/CIFS)
- SSH*
- VPN** (L2TP/IPsec)
- Xgrid

Server-side only
- Apache* (HTTP)
- FTP*
- iChat Server
- iCal Server

* While desktop versions of these servers exist on Mac OS X, they are only Kerberized on Mac OS X Server.

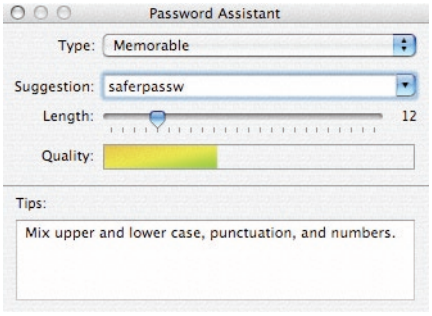
**The server component of these services is only supported on Mac OS X Server.

- **Comprehensive smart card support.** The tokend daemon works with securityd to access smart cards and similar devices, allowing them to seamlessly participate in system and application authentication activities. Token handlers are provided for the U.S. DoD Common Access Cards, MUSCLE-PC/SC, and pkcs-11. Leopard also supports the Government Smart Card Interoperability Specification (GSC-IS).
- **Advanced cryptography technology.** Mac OS X is the first system to integrate the Common Data Security Architecture (CDSA) standard for flexibly and safely managing strong cryptography (for example, AES-128), public key infrastructure (for example, OCSP, the Online Certificate Status Protocol), secure transport (for example, SSL v2/v3, TLS v1), and user interaction (for example, approving new root certificates). Mac OS X also includes the OpenSSL security library for use by legacy open source applications and supports NTLMv1/v2 for Windows compatibility.
- **Kerberos single sign-on.** Apple has adopted the MIT-developed, IETF-specified Kerberos protocol (v5) for systemwide single sign-on, allowing users to authenticate against multiple services without retyping passwords or sending them over the network. Every system automatically generates its own principals, allowing it to vend secure services that are fully compatible with other Kerberos-based implementations.

System security

Apple used its deep understanding of user experience to build security into Mac OS X from the ground up, in order to provide a safe environment with minimal inconvenience. Some of the key advantages of Mac OS X are:

- **Role-based administration.** Mac OS X does not require users to use the Administrator or root account to manage the system. Instead, the initial user (or other authorized user) can authenticate into the Administrator role, which allows one specific privileged operation at a time.
- **Network services off by default.** Unlike many systems, Mac OS X defaults to having no open ports (what is sometimes called "prehardened"). This approach avoids unnecessary exposure, yet authorized users can still enable precisely the services they need with a single click.
- **Secure launch services.** To minimize the risk of invisibly activating rogue software, the Mac OS X gatekeeper warns users before launching mail attachments, adding new Safari handlers, or opening an unknown application.
- **Fast user switching.** As a multiuser operating system, UNIX has made it possible to give users their own accounts with appropriate privileges and strict separation between them. Mac OS X makes it easy and convenient for any family to do the same, so they can easily set up individual user accounts (for example, with specific parental controls), and then quickly and safely switch between them on shared computers.
- **Password assistant.** To further reduce the risk of compromised accounts, Mac OS X warns users about choosing an easily guessed password and encourages stronger alternatives.



- **FileVault.** If you're worried about sensitive data on your laptop, enabling FileVault will encrypt all or part of your home directory using government-grade AES-128 encryption. The password will be securely stored in the user's keychain, which can only be unlocked by a personal password or a site-specific master password. In Leopard, you can even set up an encrypted swap file to avoid any possible data leakage.

- **Fast, reliable security updates.** While all network services are disabled by default, Apple is still committed to providing rapid-response software updates to address theoretical vulnerabilities identified by CERT/FIRST or other groups. The user is periodically prompted to install these updates, which with Mac OS X Server v10.5 can be limited to only those updates preapproved at their site. Administrators can also use Apple Remote Desktop to proactively push essential updates onto systems.

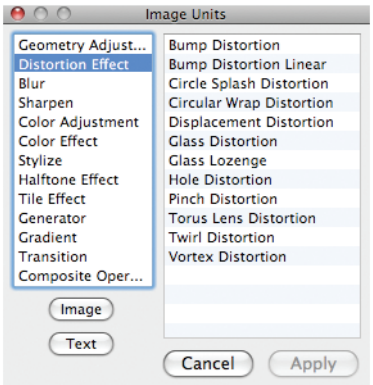
Compatible, Optimized Libraries

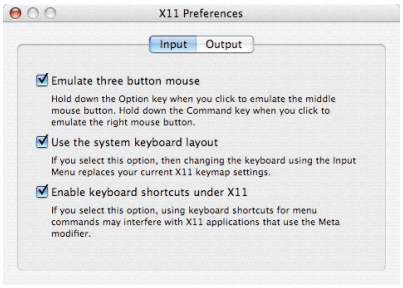
Mac OS X bundles an optimized set of traditional UNIX libraries as well as several powerful enhancements.

Application libraries

- **Standard C library.** Mac OS X C APIs now conform to the UNIX 03 specifications—SUSv3 (Single UNIX Specification, v3), XPG4 (X/Open Portability Guide 4), and POSIX (IEEE 1003.1, Edition 2004)—making it easier than ever to port UNIX and Linux code to Mac OS X. Apple also supports the usual POSIX_C_SOURCE #define, for code that requires less strict enforcement.
- **Internationalization.** Mac OS X uses a 32-bit wchar_t for maximum flexibility in representing Unicode data strings, along with GNU libiconv for converting between different character sets. The string-handling APIs also support UNIX (POSIX) locales.
- **Markup language processing.** XML/HTML processing is supported via GNOME's libxml2 and libxslt, which along with libtidy form the basis of Cocoa NSXML APIs. The KDE project's khtml and kjs similarly form the basis of Apple's open source WebCore framework and the Cocoa WebKit, which supports HTML editing and scriptable plug-ins. This technology also undergirds Safari support for RSS feeds, which works with various versions of RSS, RDF, and the Atom API.
- **SQlite.** SQlite, the embedded public domain SQL database, is available as a framework for developers, as well as being part of the Cocoa Core Data framework for data modeling and automatic persistence.
- **Apple System Logger (ASL).** This next-generation implementation of syslog is a unified, network-aware facility for storing the multitude of log files generated by applications and services in a single, queryable database, simplifying both maintenance and analysis.
- **Graphics libraries.** As with every Apple product, Mac OS X is designed for a compelling graphical experience. To that end, it provides a powerful range of toolkits built on top of Quartz and OpenGL.
- **Image processing.** Core Image and Core Video make use of the powerful graphics processing units (GPUs) on modern video cards for manipulating both static and time-based images, and together they are used for the high-level Image Units API and Core Animation effects.

Safari RSS is built on powerful open source XML and HTML libraries.





X11 integrates with the native input system, including support or emulation of multibutton mice.

X Window System

X11 for Mac OS X includes a complete, smoothly integrated X11 Window System on the Mac OS X DVD, enabling Mac OS X to run UNIX GUI applications side by side with Cocoa, Carbon, and Java applications. X11 for Mac OS X is based on X.org's X11 Release 7, the version used on Linux, BSD, and other UNIX systems. Features include:

- **Direct Quartz integration.** X11 runs directly on top of native CoreGraphics APIs and is tied directly into the native event system, for minimal overhead (reducing memory and CPU usage) and Quartz hardware acceleration (including Quartz Extreme and Exposé).
- **High-performance OpenGL.** X11 GLX applications have direct access to hardware OpenGL rendering, for the same high-speed performance as native OpenGL applications.
- **Rootless and full-screen mode.** Rootless mode allows X11 windows to run on the same desktop as native Aqua windows, making it easy to work with both at the same time. X11.app also provides a full-screen option that runs all X11 windows on a separate screen with an X11 root window in the background, using a hot key to switch back and forth.
- **Quartz window manager.** The Quartz window manager provides Aqua title bars and buttons for X11 windows, including fully functional close, minimize to Dock (using Genie or Scale), and maximize buttons. Users can optionally install and use standard X11 window managers instead.
- **Dock menu.** The menu available from the X11 Dock icon lets you view and pick any of the current X11 windows, as well as launch additional applications defined in the user-customizable Application menu, so you can easily bring up a new or existing X11 window directly from the Dock.
- **Autolaunch.** Launchd is configured to automatically launch the X11 window server whenever an X11 application is run from either the command line or the Finder.

OpenGL Utility Toolkit

OpenGL Utility Toolkit (GLUT) is a cross-platform toolkit for writing OpenGL programs that supports an Aqua compatible look and feel on Mac OS X.

Tcl/Tk

Tk/Aqua is a native Mac OS X implementation of the most popular UNIX GUI toolkit. It enables developers to write cross-platform GUI applications using Tcl or Python that will run directly under Quartz without requiring X11 to be installed or active.

WxWidgets

WxWidgets is a native port of the popular cross-platform GUI toolkit, including Python, Perl, and C++ bindings.

Numerical libraries

Leopard includes a robust suite of hand-optimized standard math libraries, plus high-performance, state-of-the-art libraries for digital signal processing and large-number operations. While the libraries work on every Macintosh, they are tuned to take advantage of native hardware capabilities, including 64-bit pointers, vector engines, and multicore CPUs.

The standard C math library in Mac OS X (libm, part of libSystem) complies with both C99 and IEEE 754. In addition to the traditional double-precision operations, it supports very fast single-precision functions, as well as long double and complex double functions via an extended match library (-l mx).

In addition, there is a dedicated Accelerate framework that provides:

- **vForce.** vForce is designed to wring optimal efficiency from modern hardware by specifying multiple operands at once, using computations rather than table lookups, and allowing only default IEEE exception handling.
- **Vectorized digital signal processing (vDSP).** Accelerate Fast Fourier Transforms (FFTs), convolutions, squares, and other common video and audio processing tasks for both single- and double-precision data.
- **Vector image processing (vImage).** Optimized routines for convolutions, compositing, and color correction, even for gigapixel images.
- **BLAS (Basic Linear Algebra Subprograms) Levels I, II, and III.** These high-quality "building block" routines perform basic vector and matrix operations using standard APIs.
- **LAPACK (Linear Algebra Package).** Written on top of BLAS, LAPACK provides industry-standard APIs for solving common linear algebra problems. Mac OS X extends the cross-platform FORTRAN and C APIs to cover both real and complex floats and doubles.
- **vMathLib.** This vectorized version of libm provides parallelized transcendental operations, enabling you to perform standard math functions on many operands at once.
- **vBigNum.** These basic arithmetic operations for manipulating large integers enable you to perform math operations on 128-bit integers, which is especially useful in applications such as number theory and cryptography.
- **OpenMPI.** In addition to Accelerate, Apple bundles the open source OpenMPI toolkit, which implements the MPI 2.0 (Message Passing Interface) standard for distributed computing. This includes dev tool support for mpicc, opal, orte, and other precompilers.

UNIX environment resources

- Mac OS X Intro for UNIX Developers: developer.apple.com/unix
- Darwin open source projects: developer.apple.com/darwin
- Mac OS Forge community site: www.macosforge.org
- Mac OS X FAQ—Mac OS X UNIX tutorial: www.osxfaq.com/Tutorials/LearningCenter

The UNIX User Experience Redefined

UNIX users will quickly recognize the full BSD command-line environment in Mac OS X, with the usual editors (for example, emacs, vim, pico/nano), utilities (ls, cp, gnutar, and so on), and shells (bash/sh, tcsh/csh, zsh, and ksh). But there's much more to Mac OS X—not only friendly applications for productivity and the digital lifestyle, but also powerful services, utilities, and tools that will improve the workflow of any command-line user.

Next-generation services

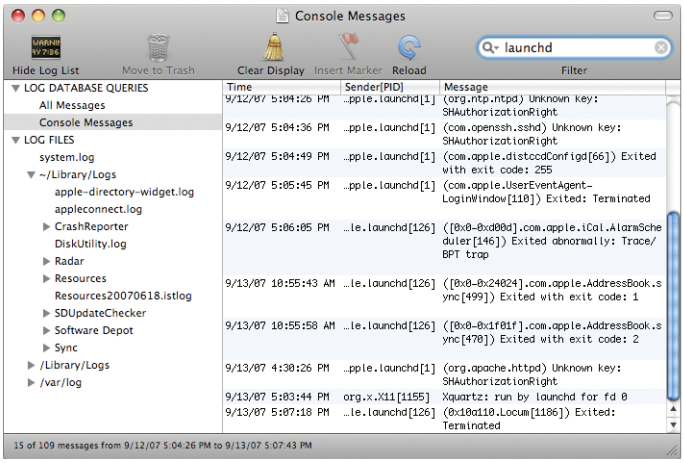
Mac OS X includes several breakthrough technologies that promise to dramatically expand the capabilities of the UNIX command line.

- **Spotlight.** Spotlight is a lightning fast way to find anything on your computer—documents, email, and so on—based on their metadata (for example, keywords and properties). The command-line tool mdfind returns a list of path names whose metadata match a given string, rather like a systemwide grep, while mdls displays metadata associated with a file. Spotlight can also remotely search mounted AFP volumes.
- **Xgrid.** Xgrid in Leopard makes it easier than ever to run virtually any command-line program (such as a scientific computation or a multimedia renderer) on an ad hoc grid of Macintosh computers. New APIs allow Xgrid to run without a preconfigured controller (for example, on a single multicore system) or characterize exactly which agents a job should run on. In addition, you can launch Xgrid jobs from OpenMPI (or vice versa).
- **Quick Looks.** Quick Looks is a new API that enables the system to quickly obtain preview images of documents for use with Time Machine, Dock icons, or the new Cover Flow view in Finder.
- **Time Machine.** Time Machine automatically maintains archival copies of all your documents, applications, and personal information on a second hard drive. Even better, it provides a beautiful, intuitive interface for discovering and recovering deleted items.

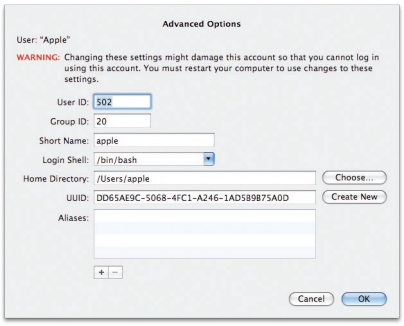
Utility applications

In addition to the dozens of rich GUI applications for end users, Mac OS X ships several advanced applications (primarily in /Applications/Utilities) of particular interest to developers, administrators, and UNIX enthusiasts, including:

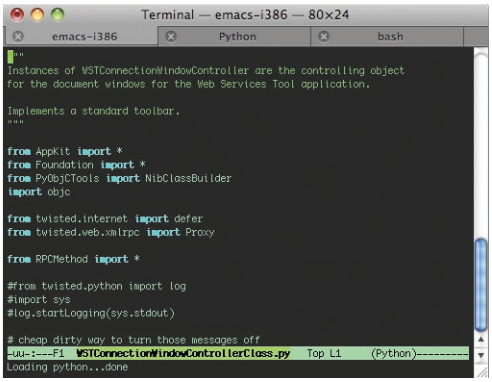
- **Activity Monitor,** which allows you to monitor the resource consumption (CPU, memory, disk, networking, and so on) of all processes on the system
- **Console,** an updated front end for searching the system log database



- **Directory,** a new application for managing shared contacts and resources, particularly for use with Mac OS X Server
- **Directory Utility,** to manage NFS exports, configure directory bindings, and manage the root password (replacing NetInfo Manager)
- **Disk Utility,** a simple graphical tool for creating/repairing volumes, partitions, and disk images
- **Grapher,** a rich environment for visualizing 2D and 3D mathematical functions
- **Keychain Access,** which lets you create and manage all your passwords and certificates
- **Network Utility,** a unified graphical front end to several common diagnostic tools, for example, netstat, ping, lookup, traceroute, and whois
- **System Preferences,** which provides a friendly GUI for managing network settings, user accounts, sharing, and other functionality that previously required manually editing configuration files
- **Terminal,** which has been completely rewritten for improved international support, adding a Safari style tabbed interface and simplified preferences for even smoother access to the UNIX command line

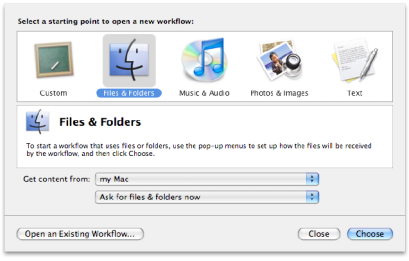


Use the Advanced Options contextual menu to configure low-level details after creating a local user account.



Scripting and automation

- **AppleScript** is Apple's native language for application automation. It's English-like syntax generates Apple events, which use a scripting dictionary (provided by most Mac applications) to programmatically create, edit, or transform their documents.
- AppleScript and other OSA **(Open Scripting Architecture)** scripts can be activated by contextual menus, user interface elements (via AppleScript Studio), iCal events, and even folder actions (on drag and drop).
- The **osacompile, osadecompile, osalang,** and **osascript** tools let you compile, describe, and execute any conforming script.
- **Automator** provides a graphical environment for assembling Actions (typically built from AppleScript or shell scripts) into sophisticated workflows, which can be saved as standalone applications or a wide range of plug-ins.
- **DTrace** trace points have been embedded in Perl, Python, and Ruby, enabling low-level analyses of applications written in those languages.
- **Cocoa bridges** (PyObjC, RubyCocoa) enable scripting languages to invoke, subclass, and be subclassed by native Objective-C objects.



New Starting Points make it easier than ever to create Automator workflows.

- The new **BridgeSupport** framework enhances bridges by providing additional meta-data and stubs for inline functions; almost every Cocoa framework in Mac OS X has complete bridge data for use by Python, Ruby, or any other BridgeSupport-enabled language.
- Python and Ruby can be easily used for GUI programming thanks to **Xcode templates** and **Interface Builder parsers.**
- The **ScriptingBridge** framework provides an elegant way for Cocoa applications (including bridged scripting languages) to generate Apple events based on an application's dictionary, even generating appropriate header files if necessary.
- Mac OS X now includes a complete **Ruby on Rails** stack, including the **Mongrel** web server, **Capistrano** deployment engine, **Ferret** search utilities, and **OpenID** authentication. Mac OS X Server also facilitates Rails deployment via a Bonjour enabled **mod_proxy_balancer.**
- Thanks to the use of **Ruby gems,** Mac OS X provides the Hpricot HTML/XML parser and Needle dependency injector, plus Ruby bindings for termios, libxml2, and sqlilte3. Also, the irb Ruby interpreter now has persistent history, syntax coloring, and full readline editing.
- **Python** has readline editing, as well as Twisted for web services, easy_install for setup, and NumPy for numerical calculations. There are also Quartz and Cocoa bindings for converting images and documents.
- **Perl** has been extended with Mac::Glue for OSA compatibility, DBI for database access, LWP for web access, MacError and mp2bug for error reporting, and ptardiff for archive comparisons.
- **PHP** has been upgraded to version 5 for use with Apache 2.0, though older versions of both are available (as an option) on Mac OS X Server.
- **swig,** the Simplified Wrapper and Interface Generator, generates script bindings from C APIs for a wide range of languages. For example, it was used to generate the Python bindings for CoreGraphics used for image conversion and PDF file generation.

Additional command-line tools
In addition, Mac OS X adds or improves a number of command-line tools—complete with man (1) pages—for the convenience of UNIX users. For example:

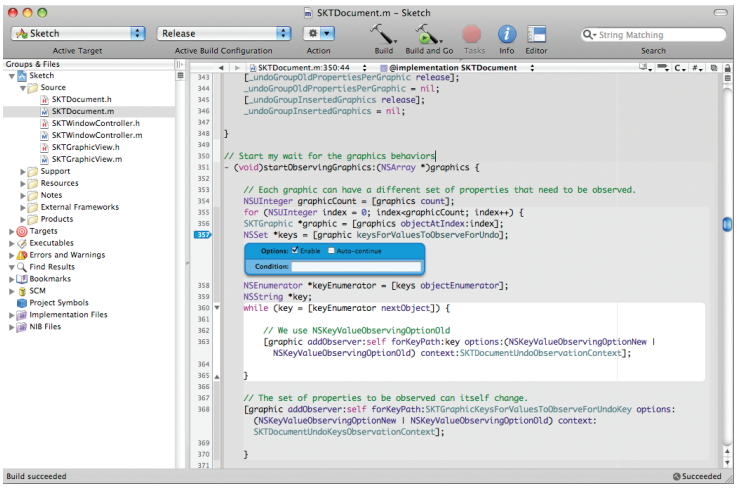
- **afconvert, afinfo, afplay:** Manage audio files.
- **applesingle, binhex, macbinary:** Encode and decode Apple double-formatted files.
- **bless, disklabel, disktool, diskutil, drutil, fsck, hdiutil, pdisk:** Create, identify, manage, and fix Mac OS X disks, file systems, and disk images.
- **codesign, codesign_allocate, csreq:** Create and manipulate code signatures.
- **cp, mv, scp, emacs, vim, pico:** Properly handle HFS+ resource forks using the extended attribute APIs, as do archivers such as tar, rsync, gzip, bzip2, and cpio.
- **createhomedir, mnthome, vifs:** Create and manage home directories and mount points.
- **defaults, plutil:** Read and modify application defaults or other property list files in ASCII, XML, or binary format. plutil now outputs to the command line (-o -) for easier integration into pipelines.
- **dscl, ds*, newgrp:** Directory services command lines to manipulate Open Directory data stores (replaces Apple's legacy NetInfo tools).
- **fs_usage, fuser, latency, sar, sc_usage, spindump:** Display various system usage statistics.

- **installer, packagemaker, softwareupdate, xar:** Create and use Mac OS X install packages.
- **ioreg & kextstat:** Show device drivers and other kernel extensions in use.
- **kadmin, kdb5_util, ktutil:** Manage Kerberos tickets, along with kinit, klist, and kdestroy, and the sso_util for single sign-on.
- **kerberosautoconfig, slapconfigkdcsetup, krbservicesetup:** Configure the Kerberos server.
- **lsm, mecab:** Linguistic analysis and latent semantic mapping.
- **mDNS, dns-sd:** Diagnostic tools for testing Bonjour service discovery.
- **nvram, pmset:** Manage open firmware and power management settings.
- **notifyutil, pcast, pubsub:** Interact with their respective services (notifyd, Podcast Producer, PubSub framework).
- **open:** Invoke launch services on an arbitrary document or application, which is equivalent to double-clicking it in the Finder.
- **path_helper:** Configure paths for the defaults shells using /etc/paths, /etc/paths.d, and /etc/manpaths.d (located in /usr/libexec)
- **pbcopy, pbpaste:** Move data between stdin/stdout and the Mac OS X pasteboard.
- **sandbox_exec:** Execute a process inside a secure "sandbox."
- **say:** Convert text to audio output or a file via Speech Synthesizer.
- **security:** A simple command-line interface for managing keychains, individual keys, and X.509 certificates.
- **sips:** Interface to Scriptable Image Processing Service for manipulating the format and color space of bitmap images (for example, rotate, scale, crop, tint, pad).
- **svn*, svk:** Subversion source code management and versioning.
- **sw_vers, uname:** Display Mac OS X version information.
- **syslog and logger:** Modern and traditional interfaces, respectively for sending, viewing, and managing system log messages.
- **tidy, xml-config, xmlcatalog, xmlwf, xpath, xslt-config, xsltproc:** Clean up and reformat XML/HTML documents.
- **tiffutil, tiff2icns:** Manipulate TIFF files for use in Mac OS X applications.
- **uuidgen:** DCE-compatible Universally Unique Identifiers.
- **xml2man:** Generate man pages created from the Man Page Generation Language (MPGL) from HeaderDoc 8.
- **zfs, zpool, fstyp* mount_union, newfs_udf:** Manage ZFS and other file systems (read-only, for testing purposes only).

Integrated developer tools

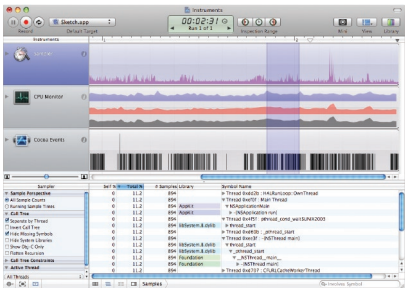
Every copy of Mac OS X ships with the same developer tools as those used by Apple engineers.

- **Xcode.** Xcode 3 is Apple's trailblazing integrated development environment (IDE) for Mac OS X. It supports a wide range of languages—including C, C++, Objective-C, and Java as well as AppleScript, Python, and Ruby—and can be used for Cocoa, Carbon, and even Darwin development.



Xcode shows your errors, warnings, and data values as you debug, all within the same editor window.

- **xcodebuild and xed.** In addition to building projects from the command line using xcodebuild, you can use xed to tell Xcode to edit specific source code.
- **agvtool, xcode-select, xcodeindex.** Manage versions, tool paths, and source indexes for Xcode projects.
- **Interface builder.** Interface Builder is a graphical editor for designing user interface components for both Carbon and Cocoa applications. Interface Builder makes creating an application's user interface easy, and it supports core animation for dynamic effects. IB files can also be validated and examined using the **ibtool** command-line utility.



Instruments displays analysis data along a timeline, highlighting cause-and-effect relationships.

- **Instruments.** This new analysis tool provides a GarageBand-like track interface for aggregating data from DTrace and other "instruments" and is usable from the command line.
- **Shark.** Shark provides a rich, source-level view of application hot spots, plus friendly advice on how to tune your application for optimum processor performance.
- **GNU Compiler Collection (GCC) 4.** Mac OS X is built using the same GCC 4 code as that hosted on the FSF public repository, allowing easy collaboration with the wider community of open source developers.
- **Dynamic linker.** dyld is the dynamic link editor for loading and linking dynamic libraries written in Apple's native MACH-O format, which for compatibility with other UNIX systems supports standard dlopen/dlclose calls and semantics.
- **Java.** Mac OS X includes the full suite of Java compilation and debugging tools (javac, rmic, java, and jdb) based on JDK 1.5.
- **IORegistryExplorer.** This application has been revised to provide beautiful diagrams of the various dependencies between IOKit components and devices.

- **Property list editor.** This allows users to easily view and edit property lists, as well as read and write XML, binary, or traditional ASCII format. Similar functionality is available with the **plutil** command-line tool, which can validate and convert among all the different formats.
- **FileMerge.** This provides a powerful graphical view of changes between files (analogous to that from the UNIX diff command) and is particularly useful when refactoring in Xcode.
- **Build tools.** Mac OS X ships with two versions of make (the default GNU make as well as BSD make), ant for Java projects, and the usual parsing tools (lex, flex, yacc, and bison).

Mac OS X Server

While Mac OS X Server has the same robust core as Mac OS X, it adds industrial-strength features required for business-critical server deployments. Designed for "headless" operation, Mac OS X Server enables you to install and configure services without needing to connect a monitor to the server. Powerful remote management tools allow you to securely manage services from anywhere on the network or over the Internet, and support for SSH provides secure access from the UNIX command line. To keep your systems up and running, Mac OS X Server has built-in tools for monitoring systems, preventing accidental shutdown, and recovering services quickly in case of network or power failure.

For More Information

Now that you've been introduced to the robust UNIX functionality of Mac OS X, here are several resources to help you find out more about specific topics.

- Mac OS X overview: www.apple.com/macosx
- UNIX features: www.apple.com/macosx/features/unix.html
- Open source projects: www.apple.com/opensource
- Scientific computing: www.apple.com/science
- Open source code releases: developer.apple.com/opensource