

# Technical Note TN2029

## Mac OS X v10.1

This Technical Note was created for application developers interested in writing software that is compatible with Mac OS X. This list includes changes that affect API level programming and product testing, it is not intended to be an exhaustive list of all the changes in this software update.

### CONTENTS

[Apple Help](#)  
[AppleScript](#)

[Scripting Additions](#)  
[Script Editor](#)  
[Script Runner](#)  
[Tools](#)  
[Finder Scripting](#)

#### [Audio](#)

[Audio HAL](#)  
[Audio Toolbox](#)  
[AudioUnit](#)  
[IOAudioFamily](#)  
[Sound Manager](#)  
[Speech Recognition Manager](#)

#### [BSD Commands](#)

#### [Carbon](#)

[Alias Manager](#)  
[Appearance Manager](#)  
[Apple Type Services \(ATS\)](#)  
[ATSUI](#)  
[Apple events](#)  
[Carbon Core](#)  
[Code Fragment Manager](#)  
[Component Manager](#)  
[Control Manager](#)  
[Date & Time Utilities](#)  
[Dialog Manager](#)  
[Drag Manager](#)  
[Event Manager](#)  
[File Manager](#)  
[Folder Manager](#)  
[Font Manager](#)  
[FontSync](#)  
[Gestalt Manager](#)  
[Help Tags](#)  
[Icon Services](#)  
[Menu Manager](#)  
[MLTE](#)  
[Navigation Services](#)  
[Process Manager](#)  
[QuickDraw](#)  
[Script Manager](#)  
[Text Encoding Converter](#)  
[Text Services Manager](#)  
[Unicode Utilities](#)  
[Window Manager](#)

#### [Classic Runtime](#)

#### [Cocoa](#)

[AppKit](#)  
[Foundation](#)  
[NSDrawer](#)  
[NSFileManager](#)  
[NSFileWrapper](#)  
[NSOpenGLPanel](#)  
[NSSavePanel](#)  
[NSScrollView](#)  
[NSSplitView](#)  
[NSString](#)

#### [CoreFoundation](#)

[CFNetwork](#)  
[CFString](#)  
[CFURL](#)

#### [Dock](#)

#### [File Systems](#)

[AFP Server](#)  
[AppleShare Client](#)  
[HFS](#)  
[HFS+](#)  
[MSDOS](#)  
[NFS](#)  
[ISO 9660](#)  
[SMB](#)  
[UDF](#)  
[WebDAV](#)

#### [Graphics](#)

[ColorSync](#)  
[Draw Sprocket](#)  
[ImageCapture](#)  
[OpenGL / OpenGL Carbon / AGL](#)  
[Printing](#)  
[Quartz 2D](#)  
[QuickTime](#)

#### [Hardware/Devices](#)

[Disk Arbitration](#)  
[IOKit](#)  
[KEXT Management](#)  
[Mass Storage](#)  
[PCCard Support](#)  
[SCSI Architecture Model \(SAM\)](#)

#### [Java](#)

[AWT](#)  
[Bridge Technology](#)  
[Embedding](#)  
[Graphics](#)  
[HotSpot](#)  
[Swing](#)

#### [Kernel](#)

[BSD Kernel](#)  
[Mach Kernel](#)  
[Launch Services](#)

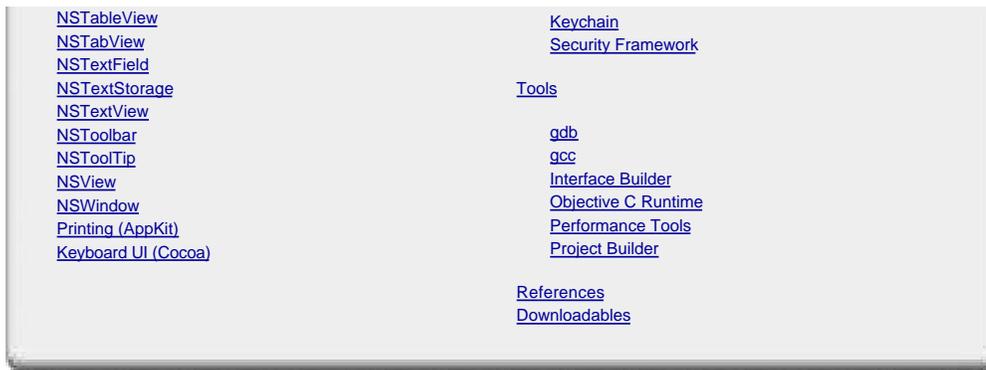
#### [LoginWindow](#)

#### [Networking](#)

[AppleTalk](#)  
[DHCP and BOOTP](#)  
[Directory Services](#)  
[Internet Config](#)  
[LDAP Plug-in](#)  
[PPP](#)  
[URL Access](#)

#### [Security](#)

[Authorization](#)  
[CSP](#)



[Oct 15 2001]

---

## Apple Help

Apple Help provides system-wide instructional help services for virtually all aspects of Mac OS. The Help Viewer is the main user interface component of the Apple Help technology.

- `AHRegisterHelpBook` was not returning an error for cases where caller of `AHRegisterHelpBook` supplied a `CFBundleHelpBookName` and no book by that name could be found in the supplied bundle (r. 2632943).
- `HREF` links in help pages referencing applications that are to be launched as a result of a click on the link now work as expected (r. 2707989).
- Apple Help Indexing Tool is now available on the Mac OS X Developer CD.

[Back to top](#)

## AppleScript

AppleScript is a component of the operating system that provides a scripting interface for users to automate actions on system and application data. Even more importantly, it allows users to access functionality of applications which would be difficult or impossible to access by hand.

- With Mac OS X version 10.1 and later, and AppleScript version 1.7 and later, Program Linking works between Mac OS X machines as well as between Mac OS 9 and Mac OS X machines. When targeting applications running on a Mac OS X machine, you must use TCP/IP networking and addressing; AppleTalk is not supported. (In addition, AppleScript 1.7 now supports encoding the login name and password into the "eppc" URL.) For a Mac OS X machine to respond to Program Linking requests, "Allow Remote Apple Events" must be enabled in the Sharing panel of System Preferences.
- AppleScript 1.7 for Mac OS X 10.1 can use Internet applications as targets of Tell blocks. The syntax is

```
tell application "http://server.domain.com/application"
  call xmlrpc {methodName:"XMLRPCMethod", parameters: {"hello", 25}}
  -- or
  call soap {methodName:"SOAPMethod", method namespace uri: "urn:xmethodsTest",
  parameters: {name:"Fred", age:32}, SOAPAction:"urn:xmethodsTest#Test"}
end tell
```

The URL for the application reference must be a correct URL for a SOAP (Simple Object Access Protocol) or XML-RPC (eXtended Markup Language Remote Procedure Call) Web server.

An alternate syntax is:

```
tell application "application" of machine "http://server.domain.com"
```

but this form will be converted to the form above when the script is compiled (r. 2412943, 2718557).

- AppleScript now defines the following file name extensions for file names:

Type of File	File Name Extension
script text	.applescript
compiled script	.scpt
scripting addition	.osax
scripting dictionary	.asdictionary

- Because many Mac OS X files use "UNIX-style" line endings (using the linefeed character, or ASCII character 10),

AppleScript versions 1.7 and later explicitly support embedding the linefeed character in strings with `\n`. Scripts that use this feature can be run on older versions of AppleScript, but if decompiled, the linefeed character will not be visible in the decompiled output (but will recompile correctly) (r. 2465565).

- Made changes to support saving in `typeFileURL`. See Technical Note [TN2022, "The Death of typeFSSpec"](#) for details (r. 2710315, 2708208).
- In AppleScript 1.5 through 1.6, a script's global variables could be lost under certain rare circumstances. If a script contains a Tell block that targets a variable, and that variable specifies an application that does not exist, the "Where is application <name>?" dialog would be presented when the script is executed. After that point, further references to global variables in the handler that included the Tell block would fail. This has been fixed in AppleScript 1.7 for Mac OS X and Classic.
- In AppleScript 1.5 and 1.6 for Mac OS X, scripting additions behaved fundamentally differently than in Mac OS 9: they were only installed in certain applications that were AppleScript-aware. If a scripting addition command (such as 'display dialog') was sent to an application that was not AppleScript-aware, the event would instead be executed in the application running the script (e.g. Script Editor); that means the dialog would come up in the Script Editor layer, not the Tell target application. In AppleScript 1.7 in Mac OS 10.1, the scripting additions mechanism has been redesigned to behave more like Mac OS 9, and scripting addition commands sent to applications will be executed in the process space and window layer of the target application.
- AppleScript 1.5 through 1.6 often launched applications unnecessarily in order to get their terminology, and often used the terminology of the script editing application (such as Script Debugger) instead of application terminology. This has been fixed in AppleScript 1.7.
- In AppleScript 1.5 through 1.6 for Mac OS X, the version property of AppleScript or an application would be displayed as raw data rather than a string (unless the version property was explicitly converted to a string). This has been fixed in AppleScript 1.7 (r. 2569693).
- In AppleScript 1.5 and 1.6 for Mac OS X, certain operations with 'file' object specifiers (e.g. 'file "Mac OS X:Applications:Terminal.app:" as alias) would fail, when similar operations would succeed on AppleScript 1.3 through 1.6 for Mac OS 9. These operations now work in AppleScript 1.7 for Mac OS X. Note that, as always, while you can pass a 'file' object specifier as a parameter to an event, or perform certain operations on it (like assignment and coercion), simply evaluating 'file "pathname"' is itself an AppleScript error.
- AppleScript 1.5 and 1.6 for Mac OS X could not target Mac OS 9 control panels running in the Classic environment. This has been fixed in AppleScript 1.7. Note that the Classic environment only supports a small number of control panels from Mac OS 9 (such as the Launcher).
- Accessing paragraphs of AppleScript strings did not work correctly in AppleScript 1.5 through 1.6 if the line endings were Windows-style (CRLF). The result would be only the first half of the text paragraphs, interspersed with an equal number of blank strings. This has been fixed in AppleScript 1.7 and later.
- Occasionally with AppleScript 1.5 and 1.6, recompiling a script would get an End of File error reading an application's terminology. This has been fixed in AppleScript 1.7.
- In AppleScript 1.6, under certain circumstances calling a handler in an applet that has not been launched could cause an "Can't continue" error. This has been fixed in AppleScript 1.7 and later.
- In AppleScript 1.6, Unicode text values stored in script properties of a saved script could not be reloaded reliably. This has been fixed in AppleScript 1.7.
- In AppleScript 1.6, Unicode text received from certain applications that included a Unicode byte-order mark would behave incorrectly; the text would appear to begin with the Euro symbol, and some operations would fail. AppleScript 1.7 handles Unicode text with byte-order marks correctly.
- In somewhat rare circumstances, comparing two references to the same file for equality can fail even if both references refer to the same file. This is true in AppleScript 1.0 through 1.6 and has been fixed in AppleScript 1.7.
- Due to a problem in AppleScript's script parser, the language formations "isn't", "doesn't", and possessives ending with "s" would all accept a vertical bar or a double-quote character in place of the "s" or "t" following the apostrophe. In AppleScript 1.7 and later, these constructions (for example, "doesn't|" or "AppleScript'" version") are correctly noted as syntax errors.
- Apple events sent by AppleScript now include an attribute(`enumConsidersAndIgnores`) to indicate the current state of "considering" and "ignoring" (r. 2708130).
- The script parser now supports Mac, Unix, and Windows line endings (r.2468789).
- Extracting a range of paragraphs from text with Windows line endings now works (r. 2671646).
- Miscellaneous performance and stability improvements (rr. 2684555,2684651, 2700558, 2660044, 2716584, 2719078).

## Scripting Additions

Scripting additions provide a mechanism for delivery of additional functionality written in other computer languages that can be used in AppleScripts.

- All disks formatted as Enhanced Macintosh File System ("HFS Plus") are able to store files with file names greater than 32 characters and stored as Unicode text. When older software accesses such files, those names are represented in an encoded form (with a file ID such as "#28AF" in the name). With AppleScript 1.6 and earlier, the results of many Standard Additions commands (like 'path to', 'info for', and 'list folder'), as well as creation and display of alias objects, would require and return encoded names. In AppleScript 1.7, long and Unicode file names in aliases and for scripting additions are represented by styled or Unicode text. Encoded file names are no longer supported.
- From AppleScript 1.1, the 'offset' scripting addition command has not worked correctly when the direct parameter is styled text. It also has not worked correctly with the newly-introduced Unicode text type. In Standard Additions 1.7 and later it produces correct results with both styled and Unicode text (r. 1197997).
- In versions of Standard Additions prior to 1.7, the 'choose URL' command's 'using editable URL' parameter was present in the Scripting Additions dictionary but was nonfunctional (URLs were always editable regardless of the setting of the parameter). It behaves correctly in Standard Additions 1.7 and later: if set to false, the user can choose a URL only by browsing, not by editing the URL in the text box (r. 2635064).

- The 'choose from list' command in Scripting Additions 1.4 through 1.6 allowed the scripter to create a situation where it would return an empty selection even if the scripter specifically disallowed that case. (This can be done by supplying a default item that does not appear in the list). In Scripting Additions 1.7 and later, the "OK" button will not be enabled in such cases (r. 2644023).
- In Mac OS X, the 'info for' scripting addition in Standard Additions 1.7 now returns additional record items that provide information about the file's extensions. The 'displayed name' item returns the name of the item as the user sees it in the Finder or file dialogs, and the 'name extension' item now returns the file extension (if any) that is on the indicated file (without the ".", which is considered a separator, and not part of either the name or the extension). Both of these fields (along with the 'name' field itself) are Unicode text values (r. 2716929).
- In Standard Additions 1.6 and earlier, the 'info for' scripting addition would fail if any piece of information in the requested file or directory was unavailable. On UFS (Mac OS X) disks and mounted NFS volumes, the file creation date is unavailable, and accessing it causes a -8850 error (in trying to convert it to Universal Time). Starting with Standard Additions 1.7, any missing piece of information in the 'info for' result is provided as the value 'missing value' without failure. In AppleScript 1.6 and earlier, magnitude comparisons (<, <=, >, >=) against the 'missing value' would signal a coercion error for most types; in 1.7 and later, these comparisons always return 'false' without error. These two changes ensure that scripts that use 'info for' against UFS and NFS volumes do not fail, though they may not get correct results because any comparison to the creation date is always false (r. 2648937).
- In Standard Additions 1.6 on both Mac OS 9 and Mac OS X, the 'info for' scripting addition would return an error -1401 for a nonexistent file, rather than the normal -35 returned in previous versions. In Standard Additions 1.7 trying to get the info for a nonexistent file (or a file in a nonexistent directory or inaccessible disk) will return the -35 error (r. 2664789).
- The 'read' command in Scripting Additions 1.6 had specific problems with the "using delimiters" parameter: in previous versions, 'read f as list using delimiters {tab}' would return a list, but in AppleScript 1.6, it returns a list of one-item lists. In AppleScript 1.7 and later it behaves as it did prior to version 1.6 (rr. 2665233, 2665169).
- The 'read' command in Scripting Additions 1.6 had specific problems with the 'before' and 'until' parameters if used in conjunction with the 'as' parameter. Instead of returning a value of a given type, it would return random data or an error. This has been fixed in Scripting Additions 1.7 and later.
- Mac OS 9 introduced the concept of a "package", that is, a file composed of a folder of several files and folders, which is treated as an integral file by the Finder and other applications. The 'info for' scripting addition in AppleScript 1.6 and earlier treated such packaged data application files as folders, not files, and would not return file- or application-related information about them. In Standard Additions 1.7 and later, getting the information for any directory (including package folders) returns an additional record item "package folder" (if this Boolean value is true, then the directory is a package), and the file's type and creator.
- Standard Additions 1.5 through 1.6 changed the way that the random number seed worked in the 'random number' scripting addition. Older versions used the system clock as the seed value if no seed value (or a seed value of 0) was provided; Scripting Additions 1.5 and 1.6 always use a seed value of 0 in these cases. In Scripting Additions 1.7 the older behavior has been restored. In addition, the seed value is now taken from the 60Hz 'ticks' counter rather than the system clock, because the increased speed of new machines makes it more likely that two consecutive random number calls may receive the same system clock time as their seed value (r. 2675978).
- In Mac OS X 10.0 through 10.0.4, the 'choose file name' scripting addition in Standard Additions 1.6 would crash if given an empty string ("") as the default file name. This has been fixed in Standard Additions 1.7. In addition, on Mac OS X the default file name can be a Unicode string.
- Fixed crash when executing script 'choose file name defaultname ""' (r. 2723566).
- 'Adding a OSAX in one domain can make AppleScript forget other OSAX domains.' Fixed now: we iterate through all the folders if any of them have changed (r. 2702081).

[Back to top](#)

## Script Editor

The Script Editor is a program installed with Mac OS that allows for creation and editing of scripts in any OSA scripting language.

- In Script Editor 1.6 for Mac OS X 10.0.0 through 10.0.4, you could not choose the Finder with the "Browse" option from the "Open Dictionary..." dialog box. This has been fixed by Standard Additions 1.7 on Mac OS 10.1.
- In Script Editor 1.6 in Mac OS X, attempting to paste Unicode text into the script window would result in an error. This has been fixed in Script Editor 1.7.
- In Script Editor 1.6, printing a script to the printer would result in random characters being printed in the place of the script file name. This has been fixed in Script Editor 1.7.
- Under certain circumstances in Mac OS 10.0.0 through 10.0.4, double-clicking generic TEXT files in the Finder would cause them to be opened by the Script Editor. This has been fixed in Script Editor 1.7, which will now be chosen to open only AppleScript-generated script text files (r. 2657598).
- Script Editor 1.7 can also open files with the file extension ".applescript" even if they have no file type.

[Back to top](#)

## Script Runner

The Script Runner is a convenient user interface gadget that allows users to run AppleScripts.

- Example Scripts have been moved to /Library/Scripts/ for simplify the user's life (r. 2700647).

[Back to top](#)

## Tools

In Mac OS X, the command line tools `osacompile` and `osascript` can be used to run AppleScripts from inside of the Terminal application.

- The `osacompile` and `osascript` tools now have an "include" option ('-i') that allows you to specify application terminology to be used when compiling the given scripts. This frees you from having to specify applications in 'Tell' blocks in the script itself. You can also use this to specify the static terminology for an application with dynamic terms, so you can prevent that application from being launched when you compile the script.
- For the `osacompile` and `osascript` command-line tools, the requirement that they be invoked with full paths (i.e. `/usr/bin/osacompile`) has been eliminated. They can now be invoked with implicit paths like any other command-line tool.
- In Mac OS X 10.0.0 through 10.0.4, the '-l' and '-d' options for the `osalang` tool did not work as documented. In Mac OS X 10.1 and later, -l prints the subtype, manufacturer, flags, and name; and -L also prints the description in parentheses after the name.
- The `osacompile` and `osascript` tools previously had a limit of 32K of input text. This limit has been lifted in Mac OS 10.1.
- The `osacompile` and `osascript` tools execute as background applications; they cannot involve any user interface. If you want to call 'display dialog', 'choose from list', or other user-interface scripting additions from a script being executed by `osascript`, you must enclose it in a Tell block targeting a running application (such as the Finder).
- "'activate" fails when called from `osascript`.' Fixed, we now ignore any error from `SetFrontProcess` (r. 2689926).

[Back to top](#)

## Finder Scripting

Finder scripting has been enhanced for Mac OS X 10.1 and is now very close to the level of scripting provided by the Mac OS 9 Finder.

- When a script sets view options (list view options or icon view options), the Finder will now set the view's display setting to "uses custom view options" so the new options will actually be used (r. 2740380).
- A problem where AppleScript terminology referencing the same file object could resolve to unique (and sometimes unusable) references has been corrected (r. 2740369).
- The script 'home' would return a value that could not be used. This has been corrected and now the value returned by 'home' is the current user's home directory (r. 2738234).
- The 'whose' clause 'every disk whose format is Mac OS Extended format' was not working correctly. Now it does (r. 2737997).
- The script "make new Finder window" was not working correctly. Now it does (r. 2734299).
- The finder now interprets "::" to mean "the parent folder" when "::" is present in a path name (r. 2722378).
- In some cases, telling the Finder to make a new alias file was not working correctly. This has been corrected (r. 2716767).
- The Finder now supports the process class allowing scripts to enumerate other processes as they could before (r. 2715201).
- Scripts requesting "every file of" a folder would crash the finder. This problem has been fixed so such inquiries no longer crash the finder (r. 2711179).
- Restart, shut down, and sleep are once again part of the terminology understood by the finder (r. 2693209).
- The Finder's implementation of text comparison in "whose" clauses was not working. This has been corrected (r. 2690467).
- It was not possible to refer to a file in a Finder script using "file" with a full path. For example, the script 'file "Mac OS X:Applications:Clock.app"' would not work. This has been corrected so such references now work as expected (r. 2689925).
- The finder now does a better job of responding to the "update item" command (r. 2686070).
- It is now possible to get the contents of the selection if the selected items are on the desktop (r. 2684258).
- Disk objects now include an "ignore privileges" property (r. 2683017).
- To provide support for the finder's file name display facilities in Mac OS X and how they work with file name extensions, new properties have been added for file objects. The read only property "displayed name" returns the name of an object as it is currently being displayed in the finder's windows. The read only property "name extension" returns the file name extension for the object. Display of file name extensions can be controlled by way of the "extension hidden" property that can be set to a boolean value (r. 2683015).

References:

- [AppleScript for Mac OS X 10.1 Web Page](#)
- [AppleScript Release Notes](#)
- [AppleScript Web Site](#)
- [AppleScript Documentation](#)
- [AppleScript Technical Notes](#)
- [AppleScript Technical Q&As](#)

[Back to top](#)

---

## Audio

This section discusses changes and new features in Mac OS X 10.1 related to audio input and output.

### Audio HAL

The Audio Hardware Abstraction Layer (HAL) represents the lowest level of access to audio devices, as well as the general characteristics of the device.

- An API for accessing physical device formats has been added (r. 2667943).
- The `kAudioDevicePropertyStreamLatency` property is now implemented (r. 2642225).
- The routine `AudioDeviceGetProperty` no longer crashes when the `outPropertyData` parameter is set to NULL (r. 2711043).
- A new property, `kAudioStreamPropertyOwningDevice = 'stdv'`, has been added so streams can now obtain a reference to the device that owns them (r. 2718528).
- Wild cards now work with `DisplayHALDeviceInfo` (r. 2720184).
- Problems where calling `AudioDeviceGetProperty` or `AudioStreamGetProperty` to ask for `kAudioDevicePropertyDeviceName` or `kAudioDevicePropertyDeviceManufacturer` would return the error code `kAudioHardwareUnknownPropertyError` even though the correct property value was being returned have been corrected (r. 2724863).
- When using the `kAudioDevicePropertyDataSourceNameForID` property to translate a source OSType to a source name with the `AudioDeviceGetProperty()` routine, it was possible that the size of the resulting C string, reported in the `mOutputDataSize` field of the `AudioValueTranslation` record, was not being reported correctly. This has been corrected (r. 2724938).
- Audio HAL now uses the IORegistry's interest notification support to detect when a device becomes unavailable (r. 2724980).
- `AudioDeviceStop` has been changed so it no longer returns until its `IOTThread` is in the stopped state (r. 2726766).
- `XThreadedDevice::Stopped()` now waits for a signal if it isn't executing in its `IOTThread` (r. 2728036).
- Support for devices with hardware pass has been implemented (r. 2510134).
- Two new properties have been added allowing callers to get and set the volume in decibels. These properties are: `kAudioDevicePropertyVolumeDecibels`, that gets and sets the volume, and `kAudioDevicePropertyVolumeRangeDecibels`, that returns the range of values valid for the device (r. 2587736).
- `AudioDeviceRead` direct buffer access is now working correctly (r. 2619532).

[Back to top](#)

### Audio Toolbox

Contains APIs for AUGraph and MusicToolbox. The AUGraph is a high-level representation of a set of AudioUnits, along with the connections between them. You can use these APIs to construct arbitrary signal paths through which audio may be processed, i.e., a modular routing system. The APIs deal with large numbers of AudioUnits and their relationships.

The MusicPlayer APIs provide the services of a sequencing toolbox. This toolbox is where events can be collected into tracks, and tracks can be copied, pasted, and looped within a sequence.

- In some cases the `MusicSequenceGetIndTrack` routine was returning `kAudioToolBoxErr_TrackIndexError` when it should not have. This has been corrected (r. 2708433).
- New accessor calls have been added for `MusicTrack AUNodes` and `MIDIEndpoints` (r. 2705509).
- New APIs have been added to allow clients to query the current connection state of the various nodes in a graph at any given time. `AUGraphGetNumberOfConnections` allows you to count connections and `AUGraphGetConnectionInfo` allows you to place inquiries about individual connections (r. 2701468).
- Problems that could occur if `AUGraphInitialize` was called to initialize a graph that had already been initialized have been corrected (r. 2701458).
- A new API has been added that allows callers to place inquiries about a graph's CPU load. `UGraphGetCPULoad` returns a double value usually between 0 and 1 (range of CPU load for LAST render cycle: 0 = not running, above 1 would indicate an overload) (r. 2701450).

References:

- [The Audio Web Pages](#)

[Back to top](#)

## AudioUnit

AudioUnits are used to generate, process, receive, or otherwise manipulate streams of audio. They are building blocks that may be used singly or connected together to form an audio signal graph, or AUGraph.

- Apple provided AudioUnits now correctly support multi-channel audio where appropriate (r. 2733575).

[Back to top](#)

## IOAudioFamily

The I/O Kit family for writing kernel audio drivers, such as USB audio and MIDI drivers.

- IOAudioFamily API has been finalized and is now available for developers to use (r. 2689466)

References:

- [I/O Kit Audio Family Documentation](#)

[Back to top](#)

## Sound Manager

The Sound Manager is the Carbon API for playing and recording sound.

- The sound manager now returns a complete list of HAL devices by way of the `SPBGetIndexedDevice` call (r. 2716220).
- The `SPBRecord` routine now works as expected when an interrupt callback is specified without a buffer or a completion routine (r. 2703549).
- The error field of the parameter block was not being set to `abortErr` before the completion routine was being called when `SPBStopRecording` was called. This has been corrected (r. 2698286).
- `SPBGetDeviceInfo` now returns two `UnsignedFixed` values for `siStereoInputGain` as documented (r. 2626723).

References:

- [Sound Manager Documentation](#)

[Back to top](#)

## Speech Recognition Manager

This manager provides speech recognition support in applications.

- Speech Recognition underwent modifications to lower the memory footprint and to increase performance (r. 2687320, 2659009).
- Speech recognition fixed to work properly with USB mono input sources (r. 2689841).
- The "listen for" AppleScript command returns wrong result what it hears noise. Fixed to not return when it hears noise, rather to continue listening until it hears one of the desired phrases (or the AE times out or is canceled) (r. 2734918).
- A problem where `SRNewLanguageObjectFromHandle` could fail when passed valid data has been corrected (r. 2658996).
- A problem preventing the compilation of AppleScripts that call the Speech Listener has been corrected (r. 2598854).

References:

- [Speech Recognition Manager Documentation](#)

[Back to top](#)

---

## BSD Commands

This section describes changes and enhancements in BSD command line tools that can be executed from inside of the terminal window.

- The `sc_usage` and `latency` commands have been updated to call the `mach_absolute_time` routine for timing

information instead of accessing the processor's TBR register directly (r. 2628734).

**IMPORTANT:**

Processes requiring high resolution processor based timing information should use the one of the high level abstractions provided by Mac OS X such as `mach_absolute_time` or `UPTime` and should never attempt to access the hardware directly.

- The command `wget` has been replaced by the command `curl`.
- The `portmap` command has been updated for better security (r. 2314762).
- The `crontab` now handles the `-e` command line option correctly (r. 2738429).
- The commands `crontab` (rr. FreeBSD-SA-01:09, 2651761), `timed` (rr. FreeBSD-SA-01:28, 2661541), `rwod` (r. FreeBSD-SA-01:29, 2661550), `fetchmail` (rr. FreeBSD-SA-01:43, 2722975, 2747320), `telnetd` (r. FreeBSD-SA-01:49, 2732197), `procmail` (r. RHSA-2001:093-03, 2730082) have been updated to close potential security holes.
- The `login` command now changes the permissions on the user's terminal device to prevent another user from either reading or writing to the terminal device (r. 2724204).
- The `rm` command has been updated to provide so that it is now more convenient to use with files that have been locked in the Finder (checking Locked in the Finder sets the BSDIMMUTABLE flag). The `rm` command now provides a warning prompt allowing users logged in as root to override the IMMUTABLE flag when they attempt to delete a locked file (r. 2690268).
- The default login script for the `tcsh` command so that it now sets the `MANPATH` environment variable to include the user's `~/man` directory (r. 2592246).
- The `top` command now processes its command line arguments as expected. Before this correction there the `top` command had difficulties recognizing combined parameters such as "`-us10`" and they had to be provided as separate parameters as in "`-u -s10`" (r. 2705129).
- The `arp` command now parses all of the other supplied command line options before acting on the "`-a`" option (r. 2711665).
- A problem in the `pax` command where it would incorrectly `chown` the target of a `symlink` rather than the `symlink` itself has been corrected (r. 2713351).
- The `ps` command now processes the "`-p`" command line option as expected (r. 2715815).
- The `bsd` generic template makefiles now include templates for creating Mach-O shared library targets (r. 2717020).
- Streamlined the execution time of the `init` command that is run as the last stage of the boot process (r. 2720339).
- The `tftpd` command has been updated or improved in the following areas (r. 2253425):
  - The "`-s`" command line option has been added. This option calls to `chroot` the `tftpd` daemon,
  - Added support for running `tftpd` command as any user,
  - Added support for file transfers of files larger than 32M
  - Formatting of syslog messages generated by the command has been cleaned up,
  - Improved bounds checking on internal buffers to prevent potential buffer overflows has been added.
- A problem in the `ftpd` command has been corrected so it no longer quits if a connection sends a PASV command to the daemon before any user has logged in to the Machine (r. 2662358).
- When performing the "`<<<`" redirection `bash1` creates a temporary file in `/tmp` with a name based on the process id, writes subsequent input out to that file, and then closes the file before re-opening it as the standard input of the command to be executed. At no stage are the results of the `creat`, `writes`, or `open` checked for an error status. This problem has been corrected so error status values for these calls are checked during the operation (r. 2695582).

[Back to top](#)

---

## Carbon

Carbon is a set of programming interfaces for use in Mac OS applications, especially those being ported from Mac OS 9 to Mac OS X. Carbon includes about 70 percent of the legacy Mac OS APIs and a number of new APIs for Mac OS X.

### Alias Manager

The Alias Manager is the part of the operating system that communicates with the file system to maintain alias records that are used to keep track of file and folder locations. The Alias Manager does not create Finder alias files; the Finder creates these files and stores alias records created by the Alias Manager in them.

- Aliases referencing NFS servers now contain sufficient information to allow those servers to be mounted correctly (r. 2709380).

- A problem where the routine `IsAliasFile` always returned the result `true` has been corrected. `IsAliasFile` now calls through to `FSIsAliasFile` (r. 2680574).

[Back to top](#)

## Appearance Manager

The Appearance Manager extends the facilities provided by the Control Manager, the Dialog Manager, the Menu Manager, and the Window Manager to provide a consistent look and feel for all on screen user interface elements in the Mac OS environment.

- The `DrawThemeTextStyle/Box` routines now cache the `ATSUTextLayout` record for improved performance (r. 2503794).
- A new tag, `kThemeVariantBaseTintTag`, was added to `Appearance.h` in list of tags supported by `GetTheme`. Authors of custom controls for MacOS X can use this tag in their drawing routines (r. 2659537).
- A new API, `CopyThemeIdentifier`, was added to `Appearance.h` that allows callers to determine which Appearance Theme is in effect (r. 2675406).
- Performance improvements have been implemented in menu item text drawing, menu drawing with special fonts, and per-application menu manager initialization (rr. 2676928, 2684067, 2695346, 2696870, 2692043).
- The routine `DrawThemeTextBox` now correctly draws the special glyphs (check mark, bullet, dash, diamond) rather than using the special glyphs font (rr. 2690202, 2695604).
- The routine `TruncateThemeText` would sometimes truncate incorrectly because it was using invalid cached information. This has now been fixed (r. 2719313).

References:

- [Appearance Manager Documentation](#)

[Back to top](#)

## Apple Type Services (ATS)

ATS provides advance typography APIs that make it possible for Mac OS applications to draw Unicode text. ATS provides both low-level services for drawing Unicode text as well as much of the high-end typographical control previously provided by QuickDraw GX.

- Changed ATS to output correct PostScript for certain non-compliant TrueType fonts (which worked with the LaserWriter 8 driver) (r. 2717188).
- A problem where `ATSTFontActivateFromFileSpecification` would fail when large numbers of fonts were installed has been corrected (r. 2632647).
- The routine `ATSTFontFindFromPostscriptName` was sometimes returning incorrect font IDs. This has been corrected (r. 2625997).
- Fonts were made available to the System and applications based on the order added (earlier ones took precedence), making it impossible for an application to use its own (private) versions of those fonts. A precedence ordering scheme has been implemented that gives application-added fonts highest priority so they will override existing fonts when appropriate (r. 2563643).
- A problem where the bottom line of an Arabic or a Persian font would display correctly on the screen but would not print correctly has been corrected. Here, the right end of the line was being cut off. Now, the distribution and location of Kashidas and white spaces are more now accurate in big slope lines (r. 2533850).
- Added support for 32-bit cmap formats. Specifically, formats 8, 10 and 12, which allow mapping of additional Unicode characters (r. 2491112).

References:

- [ATS Types Developer Reference](#)

[Back to top](#)

## ATSUI (Apple Type Services for Unicode Imaging)

ATSUI extends the QuickDraw API and makes it possible for Mac OS applications to draw Unicode text. ATSUI provides both basic services for drawing Unicode text and much of the high-end typographical control formerly provided by QuickDraw GX.

- ATSUI now allows you to create separate font fallback objects which can be associated with an `ATSUTextLayout`. These objects cache the result of the font fallback process for greater performance.
- Font variations were not working correctly when imaging with CoreGraphics. ATSUI now passes the variations attribute to CoreGraphics when drawing text (r. 2570218).
- A problem where ATSUI text drawn to the printer was being printed rotated 180 degrees relative to the same text drawn to the screen has been corrected (r. 2731716).
- Many aspects of ATSUI performance have been greatly improved. Font substitution is now much faster, and overall layout performance is faster. ATSUI caches many more values and data structures to speed up text drawing performance

(rr. 2726410, 2716871, 2716169, 2715296, 2713179, 2705998, 2699489, 2673744, 2671234, 2572090, 2499393, 2487786).

- The routines `ATSUSetTextPointerLocation` and `ATSUSetTextHandleLocation` have been updated so they now support the constants `kATSUFromTextBeginning` and `kATSUToTextEnd` in the `iTextOffset` and `iTextLength` parameters (r. 2715359).
- A problem where ATSUI truncation could go past the end of a text boundary when using Lucida Grande font at 18 point has been corrected (r. 2706551).
- ATSUI was reporting bad descender values for certain fonts. Here, the value `descender+leading` was being returned as the descender value. Now, ATSUI clients that do not want the leading value to be included in this calculation can specify the `kATSULineIgnoreFontLeading` flag in the `kATSULineLayoutOptionsTag` attribute's `ATSULineLayoutOptions` value. This works nicely for fonts that have the leading value defined - when necessary, ATSUI will get leading values from `GetGlyphVectorSpecs` for fonts that do not have a leading value defined (r. 2686163).
- When changing Type 1 fonts, WorldText complains that some chars cannot be displayed in the selected font even though the characters are simple low ASCII Roman. Prevent ATSUI from displaying an error message for fonts that do not have the correct glyph assigned for U+000D (r. 2590963).
- ATSUI is now significantly more thread safe: `ATSUStyle` and `ATSUFontFallbacks` objects are thread safe in that you can share them among different threads, but `ATSUTextLayout` objects are not thread safe and should not be shared.
- ATSUI now supports Unicode surrogate pairs directly, and can handle cmap formats which support them.

References:

- [ATSUI Developer Documentation](#)
- [The ATSUI SDK](#)
- [Unicode Consortium's web site](#)

[Back to top](#)

## Apple events

Apple events provide a simple interapplication and intraapplication communications facility for Mac OS X applications.

- It is now possible to target processes on remote machines as recipients of Apple events.
- The Apple event data coercion handler `typeChar->typeLongInteger` sometimes returned a long integer on an error. We now correctly return `errAECoeercionFail` for this coercion error (r.2674451).
- Hex data output by `AEPrintDescToHandle` was being truncated at 32 bytes. This limit has been removed - hex data is no longer truncated (r. 2672759).

References:

- [Apple Event Manager Documentation](#)

[Back to top](#)

## Carbon Core

The Carbon Core incorporates essential services that are generally devoid of a user interface. These facilities range from memory management to process management.

- `CSCopyMachineName` now returns the machine's AppleTalk name (In 10.0.x `CSCopyMachineName` returned the machine's hostname).
- The routine `Delay` now returns a value `finalTicks` (r. 2736558).
- The routine `CFDistributedNotifications` can now be called outside of login sessions (r. 2698120).
- The routine `CSCopyUserName` can now be called from Carbon CFM applications (r. 2666981).

References:

- [Carbon Core Release Notes](#)

[Back to top](#)

## Code Fragment Manager

The Code Fragment Manager (CFM) is responsible for loading and preparing PowerPC executables. Almost all of the PowerPC Mac OS and all PowerPC applications depend on the services of the CFM in some way.

- The `LaunchCFMApp` command is now pre-bound against all Carbon module. As a result, less launch time binding is required for CFM Carbon applications resulting in significantly faster application launch times (r. 2699026).
- A problem with CFM reference counting on connection IDs has been corrected (r. 2696728).

- Additional work has been done to improve the performance of the Code FragmentManager's handling of file system calls when loading libraries (r. 2712553).

References:

- [Code Fragment Manager Documentation](#)

[Back to top](#)

## Component Manager

The Component Manager provides facilities for tracking and calling code modules that provide a defined set of services to one or more clients (components).

- New APIs to support two-level name space lookups and improve symbol lookup performance have been added (2689833, 2701673).
- The performance and reliability of the `GetComponentPublicResourceList` routine has been improved (r. 2713803).
- QuickTime package installations are now recognized by the Component Manager without having to restart the system (r. 2718051).

References:

- [Component Manager Documentation](#)

[Back to top](#)

## Control Manager

The Control Manager provides facilities for drawing and processing user interaction with screen items inside of windows.

- A problem where `TrackControl` was returning the value `kControlMenuPart` rather than the value `kControlLabelPart` after tracking a click in a popup menu control where the user had chosen a menu item has been corrected (r. 2427323).
- Icon control creation no longer fails if you point it at an invalid icon (r. 2713860).
- `CountSubControls` now returns 0 for popup-with-checkbox controls. Before it was returning 1 (r. 2360729).
- A new tag, `kControlTabImageContentTag`, has been defined in `ControlDefinitions.h` that allows callers to get and set icons in tab controls (r. 2447692).
- Standard controls which were returning `kControlWantsIdle` now use their own carbon event timers for their idle operations (r. 2464967).
- It is now possible to provide a NULL pointer for the second parameter to `CreateRootControl` when your application does not require that the root control be returned. Previously, `CreateRootControl` would return `paramErr` if NULL was passed as the second parameter (r. 2666166).
- A new tag, `kControlEditTextPasswordCFStringTag`, has been defined in `ControlDefinitions.h` that can be used in Mac OS X applications. This tag provides the same functionality as the non-CFString tag `kControlEditTextPasswordTag`, only it allows callers to obtain the password text as a CFString (r. 2722976).
- Data browser control drawing performance has been improved (r. 2497529, 2658680, 2601622, 2547537, 2597258).
- A problem in the Data Browser control where `SetDataBrowserItemDataText` would call `CFRetain` one additional time for the CFString value passed to it (and thereby cause a memory leak) has been corrected (r. 2706654).
- In the Data Browser control, while an in place editing command was in progress, the `EnableDataBrowserEditCommand` routine was not working correctly. This has been corrected so the `EnableDataBrowserEditCommand` routine works as expected (r. 2691325).

References:

- [Control Manager Documentation](#)

[Back to top](#)

## Date & Time Utilities

The Date & Time Utilities provide facilities for accessing the clock chip and APIs for converting dates between various formats.

- The `InitDateCache` now performs better internal error checking when reading resources it requires (r. 2718176).

References:

- [Date, Time, and Measurement Utilities Documentation](#)

[Back to top](#)

## Dialog Manager

The Dialog Manager manages user interactions with windows containing controls arranged and positioned using dialog layout templates.

- The Dialog Manager routine `SetDialogDefaultItem` now calls the Control Manager routine `SetWindowDefaultButton` to set a dialog's default button (r. 2635614).
- The routine `GetDialogItemAsControl` was returning a bogus `ControlRef` when the item was of type `kHelpDialogItem`. `GetDialogItemAsControl` now returns an error for items of type `kHelpDialogItem` (r. 2413887).
- The `StandardAlert` routine no longer forces the current application to the front since the Notification Manager (a la Mac OS 9) is now available again on Mac OS X 10.1. Applications should use the Notification Manager or `AEInteractWithUser` before posting an alert from the background, as has always been the case on classic Mac OS (r. 2713623).

References:

- [Dialog Manager Documentation](#)

[Back to top](#)

## Drag Manager

The Drag Manager supports drag-and-drop operations between windows and applications.

- `SetDragImage` no longer crashes when passed a zero-sized region. In these cases, the image will be drawn full (r. 2643332, 2700741).
- A new flavor flag, `flavorDataPromised`, has been added to `Drag.h`. This flag allows callers to mark those flavors that are promised so recipients can determine which of the flavors provided in a drag will require additional processing when their data is requested (r. 2703450).
- Data for promised flavors can now be requested during drag tracking by tracking applications and not just at receiving time. Applications retrieving promised data while tracking drags should do so with discretion.

References:

- [Drag Manager Documentation](#)

[Back to top](#)

## Event Manager

The Event Manager manages the delivery of system and user interaction events to applications.

- "Clear" key no longer triggers `IsUserCancelEventRef` to return true (r. 2726545).
- A problem where `RemoveEventHandler` could corrupt memory when called twice to remove the same handler has been corrected (r. 2719106).
- The screen snapshot key combinations `command-shift-3` and `command-shift-4` have been implemented for Mac OS X 10.1. Screenshots created using these commands are stored as TIFF files in the `-/Desktop` folder (r. 2709557).
- `FlushEvents` now properly flushes events regardless of whether an application has entered its event loop or not (r. 2667246).
- It is now possible for Carbon Event handlers to dispose of their target object while executing. The Carbon Event dispatcher no longer makes any assumptions about the validity of the handler or its target after a handler has returned. This would be useful in some cases such as a handler that disposes of its window in response to some user activity (r. 2648390).
- A new routine `GetCFRunLoopFromEventLoop` has been added to allow an application to obtain the `CFRunLoopSource`. This new facility can be used to add new `CFRunLoopSources` (sockets, pipe's) to the run loop (r. 2605077).

References:

- [Carbon Event Manager Documentation](#)
- [Event Manager Documentation](#)

[Back to top](#)

## File Manager

The File Manager provides services for storing and retrieving disk-based information.

- Carbon was caching of the EOF value in the Carbon FCB to avoid having to call `lseek` every time the current EOF value is required. This caused synchronization problems when multiple applications accessed a single file. The EOF value is no longer cached (r. 2656449).
- In Mac OS X 10.0.x, the `filesystemID` and `sigWord` in `volumeinfo` parameter blocks for Audio CDs were incorrect (in fact, they were swapped from what they are on OS 9). They are now correct in 10.1. The correct values are:

```
filesystemID = 0x4A48
sigWord = kHFSSigWord ('BD')
```

- Mac OS X, Carbon, may not have been updating the volume encoding bitmap correctly if the system was booted in one script but another script was used to create a file. This caused the Unicode->Pascal conversion to behave incorrectly. This bug caused a number of international related bugs in which files were not found, etc (r. 2739253).
- A file can now be opened for shared read/write access by the same application (r. 2725021).
- `FNSubscribe` API was added to Mac OS X (Mac OS X 10.1) to allow applications to be notified when a folder contents have been changed. The `FNSubscribe` routine is documented in `Files.h` (r. 2706955).
- If non unicode, non UTF characters are found on UFS/NFS volumes, they are now returned as '?' characters (r. 2690819).
- The routine `PBReadSync` was not returning `eofErr` if the new line bit was set. This caused `noErr` to be returned with zero bytes of data when attempts were made to read past the end of a file (r. 2677789).
- The Carbon File System now returns a `badNameErr` if the volume name of a UFS file or volume is greater than 255 bytes (r. 2648740).

References:

- [File Manager Documentation](#)

[Back to top](#)

## Folder Manager

The Folder Manager provides facilities for locating "special" folders (for example, the Extensions folder) without relying on the names of those folders. This aids developers in application localization.

- With the release of the Directory Services, 2 new `FindFolder` constants have been defined to enable developers to locate specific folders: `kDirectoryServicesFolderType` -> `*/Library/DirectoryServices`, and `kDirectoryServicesPlugInsFolderType` -> `*/Library/DirectoryServices/PlugIns` (r. 2609405).
- `FindFolder` (and `FSFindFolder`) was not taking in account changes of user id (through `setuid`). This has now been fixed (rr. 2611363, 2693365).
- A small memory leak in `FindFolder` has been corrected (r. 2644507).
- Some file systems (ie. HFS+) are case insensitive whereas some others (ie. UFS) are not. `FindFolder` now takes this into account when searching for folders (r. 2645592).
- The Folder Manager was mishandling permission privileges and was not thread safe. This has been fixed (r. 2658862).
- A new `FindFolder` constant was added to enable developers to locate the Receipts folder: `kInstallerReceiptsFolderType` -> `*/Library/Receipts` (r. 2662562).
- A new `FindFolder` constant was added to enable developers to locate the Authentication Modules Plug-ins folder: `kAppleShareAuthenticationFolderType` -> `*/Library/Filesystems/AppleShare/Authentication` (r. 2668964).
- A new `FindFolder` constant was added to enable developers to locate the MIDI Drivers folder: `kMIDIDriversFolderType` -> `*/Library/Audio/MIDI Drivers` (r. 2678403).
- Calls to `FSFindFolder(kLocalDomain, kDesktopPicturesFolderType, ...)` were returning a reference to `/Library/Appearance/Desktop Pictures/` rather than `/Library/Desktop Pictures/`. This has now been fixed (r. 2730236).

References:

- [Folder Manager Documentation](#)

[Back to top](#)

## Font Manager

The Font Manager provides system-wide services for the retrieval of glyphs for display of textual information.

- Shadow/outline width tab values from `FontMetrics` in Mac OS X 10.1 have been corrected so they are now consistent with the same values returned in Mac OS 9 (r. 2698733).

- Fract enabled bit is now respected for Type 1 fonts (r. 2668615, 2668622).
- Accessor functions have been added for the font container and font family resources (r. 2717955).
- Conversion functions for `ATSTFontRef` <-> `FMFont` and `ATSTFontFamilyRef` <-> `FMFontFamily` have been added (r. 2717952).

References:

- [Font Manager Documentation](#)

[Back to top](#)

## FontSync

FontSync provides a common method for identifying fonts based upon the content of the font rather than just the name as a document is moved from machine to machine.

- FontSync now supports `FSRefs` in addition to `FSSpecs`. This allows for using Unicode in the name of the profile as well as increasing the number of characters allowed in file names (r. 2724642).
- AppleScript support for FontSync has been added.

References:

- [FontSync Documentation](#)

[Back to top](#)

## Gestalt Manager

The Gestalt Manager provides a simple, efficient way for your application to determine the user's hardware and software configuration at runtime so that you can fully exploit the available features or inform the user if a necessary feature is missing.

- Calls to the `NewGestalt` routine when the selector was already installed were mistakenly returning `noErr`. `NewGestalt` has been corrected so it now returns `gestaltDupSelectorErr` for selectors that have already been defined (r. 2674422).

References:

- [Gestalt Manager Documentation](#)

[Back to top](#)

## Help Tags

Help Tags are the primary method for providing context sensitive help in Mac OS X. They are the Mac OS X replacement for the older Balloon Help technology.

- A new API has been added, `HMHideTag`, that allows applications to hide the tag that is currently being displayed (r. 2654097).

[Back to top](#)

## Icon Services

Icon Services provides fast and efficient facilities for retrieval of appropriate icon information used to represent files, directories, or other commonly used icons (for example, the caution alert icon, the note alert icon, and the help icon).

- New Alert icons for Stop, Note, and Caution now available (r. 2694601).
- `SetSuiteLabel` will now return `paramErr` if called with a value greater than 7 or less than 0 (r. 2388249).
- A problem causing intermittent crashes when using `PlotIconRef` to draw application icons (16x16) in a palette window has been fixed (r. 2744424).

References:

- [Icon Services and Utilities Documentation](#)

[Back to top](#)

## Menu Manager

The Menu Manager is the part of the operating system responsible for both drawing the menu bar, and drawing menus and pop-up menus on the screen while the mouse is being held down.

- New documentation for enabling Services for Carbon applications is available at the reference given below (r. 2761323).
- Contextual Menu plugins now work correctly (r. 2744024).

References:

- [Menu Manager Documentation](#)

[Back to top](#)

### Multilingual Text Editor (MLTE)

The Multilingual Text Editor (MLTE) supplies a C language interface for creating and editing Unicode text documents. Although MLTE may appear to be functionally similar to TextEdit in many ways, MLTE provides many features that are not provided in TextEdit.

- On 10.0 MLTE allowed developers to automatically handle Text InputCarbon events. On 10.1, MLTE handles a wider number of Carbon events automatically.

MLTE no longer uses Apple events for text input. MLTE now uses Carbon Text Input events. This avoids possible conflicts with other potential Apple event handlers for text input that a developer might use in their application, and it brings MLTE in line with Apple's new event handling model. (r. 2661191, 2685845).

- On 10.1, MLTE uses by default CoreGraphics for imaging. New masks were introduced for developers to select the imaging environment. For example, clients can specify the frame option `kTXNUseQDforImagingMask` to use QuickDraw Text (r. 2681781).
- The new APIs, `TXNGetSpecifiedChangeCount` and `TXNClearSpecifiedChangeCount`, have been introduced that allow clients to determine if the contents of MLTE objects have changed. These APIs are described as follows:

`TXNGetSpecifiedChangeCount` will return the number of changes for the given option mask(s). Every input action will be incremented, including undo and key press. The purpose of this API is to provide a more expandable version of existing API `TXNGetChangeCount` and allow clients to obtain various count information.

`TXNClearSpecifiedChangeCount` will allow client to reset the specified counter. Writing the document to the disk will not reset the counter automatically (r. 2664957).

- A new flag, `kTXNDisableDragAndDropMask`, that allows applications to turn off Drag and Drop has been added (r. 2744110).
- The `TXNEchoMode` routine had a potential crashing bug with certain `UniChar` values when called with the `kTXNUnicodeEncoding` parameter. This has been fixed (r. 2692659).
- MLTE now provides finer control over the view rectangle and destination rectangle. The API `TXNGetRectBounds` has been added to the set of APIs exported by MLTE that allows developers to have finer control over the view rectangle and destination rectangle (r. 2689977).
- Previously MLTE would not supply correct script info when placing non-Roman text on the public scrap in some specific cases (for example: when the non-Roman text in an MLTE object was not set to the correct font/script combination.) This problem was fixed (r. 2679828).
- `TXNGetDataEncoded` API might have failed for certain specific offset parameters (end offset < start offset \* 2) when translating from Unicode to plain text encoding. This was fixed. (r. 2661305).
- MLTE added a flag to specify that a `TXNObject` only allow a single line of text. This turns off vertical scrolling and disallows input of any linebreak characters. (r. 2661126).

References:

- [Multilingual Text Engine Documentation](#)

[Back to top](#)

### Navigation Services

Navigation Services provides new Open and Save dialogs, allowing users to locate and select files and other resources.

- Custom event handling callback routines will now receive `HighLevelEvent` notifications. Applications can use these notifications to either call `AEPProcessAppleEvent` to dispatch the event to their own handlers (r. 2719014).
- In Mac OS X 10.0.0 through Mac OS X 10.0.4 `NavChooseFolder` was not returning the correct item number in `itemHit` for a `keyDown` event directed to a custom edit control. This has been corrected (r. 2719835).

References:

- [Navigation Services Documentation](#)

[Back to top](#)

## Process Manager

The Process Manager provides a set of APIs allowing processes to find out information about other processes registered with the Process Manager.

- Speed of the `GetProcessInformation` call has been improved (r. 2651002).
- Two new APIs, `IsProcessVisible` and `ShowHideProcess`, have been defined in `Processes.h`. These routines are available for Mac OS X applications. For more information, read the descriptions in `Processes.h` (r. 2489238).

References:

- [Process Manager Documentation](#)

[Back to top](#)

## QuickDraw

QuickDraw is a part of the Mac OS used for drawing and displaying graphical information on the screen and other raster devices.

- Fixed a bug that occurred when copying from a window's port's `PixMap` to another `PixMap`; afterwards, the source port's `PixMap`'s base address would be reset to the base address of the current `GDevice` - now it's correctly reset to the base address of the main `GDevice` (r. 2756169).
- The stability of the `GetPictInfo` routine has been improved (r. 2762314).
- A problem where importing an image for which no codec exists could cause a crash has been fixed (r. 2660484).
- The routines `CreateQDContextForCollapsedWindowDockTile` and `ReleaseQDContextForCollapsedWindowDockTile` can now be called from Carbon CFM applications (r. 2676885).
- Performance of line drawing for has been improved (rr. 2438552, 2668363).
- Accessor functions for the `txSize`, `txFont`, `txFace`, and `txModeGrafPort` fields have been added (r. 2440918).
- Painting with a single color `PixPat` created using `MakeRGBPat` now works as expected (r. 2563999).
- Information returned by the routines `GetPictInfo` and `GetPixMapInfo` is accurate. Some of the fields were not accurate in previous Mac OS X implementations of these routines (rr. 2567490, 2570346).
- Problems rendering grey underlines in `TextEdit` fields under millions of colors have been fixed (rr. 2648635, 2689331).
- `QDAddRectToDirtyRegion` and `QDAddRegionToDirtyRegion` have been can now be called by Carbon CFM applications (r. 2662034).
- In Mac OS X 10.0.x, calling `DrawPicture` to draw a picture containing a `DirectPix` opcode with `packtype == 2` (stripped alpha channel) could cause a crash. This has been corrected (r. 2663619).
- A new accessor, `SwapPortPicSaveHandle`, for swapping the `GrafPort`'s `picSave` handle has been added to `QuickDraw.h` (r. 2670340).
- Calling `QDFlushPortBuffer` after a call to `CGContextSynchronize` now behaves as expected (r. 2688434).
- Support for intermixing QuickDraw and CoreGraphics calls during printing operations has been added (rr. 2690777, 2705621).
- QuickDraw now passes all `SetCursor` calls through to CoreGraphics. This allows applications to set the cursor to a custom cursor during, say, drag and drop tracking while their application is not necessarily in the foreground (r. 2693703).
- A problem where `PICT`s created using `OpenCPicture` and `CopyBits` in Mac OS X 10.0.x were not setting the `pmVersion` field correctly has been corrected. This problem prevented these pictures from being opened in some third party software (r. 2705390).
- scaling quality of the `CopyBits` and `CopyMask` routines has been improved (r. 2715736).
- Image quality problems with drawing `Pictures` containing `BitMaps` using `QDPictDrawToCGContext` have been corrected (r. 2730941).

References:

- [QuickDraw Documentation](#)

[Back to top](#)

## Script Manager

The Script Manager enables internationalization of non-Unicode applications.

- The MacCyrillic and MacCentralEuropean script bundles were added to the system. They are normally inactive unless a font from the appropriate ID range is present (e.g., Charcoal CY or Charcoal CE).

References:

- [Script Manager Documentation](#)

[Back to top](#)

### Text Encoding Converter

The Text Encoding Converter enables the conversion of text data from one encoding to another (e.g., Mac OS Roman to Unicode).

- A problem where encoding conversion between MacJapanese and decomposed form Unicode could fail has been corrected (r. 2723355).
- Support for the JIS X 0213, HK-SCS, and GB 18030 character sets has been added.
- Support for Unicode 3.1 and Unicode 3.2 (draft) has been added.
- The MacKorean/Unicode mapping was significantly revised to take advantage of Unicode 3.2, and to make it complete (approximately 140 characters were not being mapped in prior versions).
- The mapping for Symbol was revised to take advantage of Unicode 3.2.
- The UTF-8 converters now check for malformed UTF-8 more strictly, per the policy of the Unicode Consortium.
- The performance of TEC initialization was significantly improved.

References:

- [Text Encoding Conversion Manager Documentation](#)

[Back to top](#)

### Text Services Manager

The Text Services Manager provides facilities for applications to communicate with various textprocessing utilities that provide services such as special text input methods, spell checking, hyphenation, etc.

- Keyboard script switching now behaves correctly (rr. 2650765, 2688646).
- The 'International Preferences' panel of the Keyboard system preference pane now correctly shows Roman keyboard layouts when a Roman input method is installed (r. 2658289).

References:

- [Text Services Manager Documentation](#)

[Back to top](#)

### Unicode Utilities

The Unicode Utilities enable internationalization of Unicode applications.

- Fixed a bug where having a precomposed fraction character (e.g. 1/2, 3/4) with the numeric sorting option turned on would cause collation to hang.
- Added support for Unicode 3.1 and 3.2, including surrogate pairs.

References:

- [Unicode Utilities Documentation](#)

[Back to top](#)

### Window Manager

The Window Manager provides facilities for drawing and maintaining windows on the screen.

- A new routine, `IsWindowLatentVisible`, has been added to `Windows.h`.

Mac OS X applications can call this routine to determine if a window is visible on screen and also whether it is latently visible but not currently on screen.

All windows are either on screen or off screen. A window that is off screen may still be latently visible: this occurs, for example, when a floating window is hidden as an application is suspended. The floating window is not visible on screen, but it is latently visible and is only hidden due to the suspended state of the application: when the application becomes active again, the floating window will be placed back on screen (r. 2754595).

References:

- [Window Manager Documentation](#)

[Back to top](#)

---

## Classic Runtime

The Mac OS X Classic Runtime environment provides a Mac OS 9 compatibility runtime environment for legacy applications and software.

- Applications on ISO 9660 disks can now be launched in Classic (r. 2684479).
- A problem where the classic Resource Manager cache was not being invalidated after a resource file was closed has been corrected. This would only be a problem if the file refnum was reused, that is only likely to happen when the FCB table was getting full. If that occurred, then the wrong resource data for resources smaller than 512 bytes could be returned (r. 2710761).
- A number of corrections related to sleep in the Classic environment have been corrected (rr. 2705529, 2707041, 2710201, 2625336, 2653690, 2654232, 2654978, 2659472, 2663493, 2669710, 2700087, 2705234).
- MP Tasks running in the classic environment on multi-processor machines were only running on one processor. This has been corrected (r. 2663329).
- `MPLibraryIsLoaded` was returning false in the Classic environment on single processor machines. This has been changed so now the `MPLibraryIsLoaded` routine returns true on both single and multiple processor machines when called from classic (r. 2661330).
- Multiprocessing library 2.0 is now supported in classic (with the exception of the `MPTrowException` routine) (r. 2351584).
- Classic setup preferences in the System Preferences dialog now does a better job of identifying and enabling eligible classic system folders (r. 2717173).
- The Rebuild Desktop button in the classic setup preferences in the System Preferences dialog would fail to rebuild the desktop file when clicked. This problem has been corrected (r. 2703373).
- Classic applications expecting to change the cursor position by adjusting the low memory cursor position values now work as expected (r. 2729368).

[Back to top](#)

---

## Cocoa

A set of object-oriented frameworks that support rapid development and high productivity, using a dynamic runtime and rich object hierarchy.

### AppKit

AppKit is a framework that provides graphical objects with default behaviors for rapid application development, including windows, panels, buttons, menus, scrollers, text fields, and more.

- `Credits.rtf` is now supported in the "About..." dialog, along with the other supported formats (`rtf`, `html`) - just put the file in the Resources folder in your application's bundle (r. 2704891). See the discussion of the `Credits.html` file in the Standard About Panel section of the [Mac OS X 10.1 Release Notes](#) for more information about using this file.
- Drawing strings off the left edge, and through affine transforms now display correctly (r. 2632822).
- Cocoa applications crashed in very specific cases involving Unicode Hex input and the user changing the font size. These cases have been corrected (r. 2735052).
- An infinite loop could occur when 32-bit cmap tries to map a two Unicode character into a single glyph. We now pad with a null glyph to prevent this hang from occurring (r. 2734557).
- Cocoa applications can now set the contents of their Dock menu (r. 2711768).
- `NSTextView` sometimes crashed after calling `-[NSLayoutManager rectArrayForCharacterRange:...]`. We now properly check the return value (r. 2663005).

References:

- [Cocoa Documentation](#)
- [Application Kit Documentation](#)
- [Application Kit Release Notes](#)
- [Legacy Application Kit Release Notes](#)

[Back to top](#)

## Foundation

Foundation is a framework that provides the base layer of (typically non-GUI) functionality that Cocoa applications need, including classes for dealing with strings, arrays, the file system, memory management, and more.

- `NSURL` and `NSURLHandle` have been extended to access "https:" style URLs
- `NSFileManager`'s `removeFileAtPath:` now removes ".\_" catalog files for directories on file systems that don't understand resource forks, for binaries compiled against 10.1 (r. 2746605).

### Note:

Binaries compiled under 10.0.x when run under 10.1 will exhibit their old 10.0.x behavior (i.e. should they request a `directoryContentsAtPath` they'll get back ".\_" catalog/resource files) even though they're running under 10.1. Also, the existence of a preference to get 10.0.x applications to exhibit 10.1 behavior.

- `NSFileManager`'s `directoryContents` method now accounts for ".\_" catalog files, when called from binaries compiled against 10.1 (r. 2718226, see note above).
- `NSFileManager` now properly copies or moves resource forks and other HFS metadata (r. 2718221).
- `NSDictionary`s are now actually immutable when possible, instead of being really `NSMutableDictionary`s underneath (r. 2683641).
- A problem that could cause a crash in `NSSocketPort` has been fixed (r. 2650498).
- A problem that could cause a crash in `NSTimeZone`'s `+timeZoneWithName:` with Africa/Abidjan has been fixed (r. 2623966).
- A potential race condition in `NSThread` has been fixed (r. 2747448).
- A problem in the exception handling code for `-[NSConnection sendInvocation:target:]` where a variable was being released too many times has been corrected (r. 2710164).
- `NSFoundationVersionNumber` is now declared in `NSObjCRuntime.h`. Applications with compile time dependencies on particular versions of foundation can compare against this value in their preprocessor code (r. 2695088, 2695627).
- A problem where `NSData` objects created from `CFObjects` could hash their data differently than ordinary `NSData` objects has been corrected. This prevents `-isEqual:` from failing on objects that are really equal but were hashed differently (r. 2682823).
- `-[NSMutableArray removeObjectsinArray:]` now uses `-isEqual:` for comparisons rather than pointer equality (r. 2675482).
- A potential crashing problem that could occur in `NSAttributedString`s in multi-threaded applications has been corrected (r. 2617551).
- Problems found with putting spaces in the format string of a formatter passed to `NSNumberFormatter` have been corrected (r. 2566070).
- Collections now respect the initial size hint provided to them in the various initialization methods (r. 2543041).
- A case where `NSFileHandle` could leak a file descriptor has been fixed (r. 2272814).

References:

- [Foundation Release Notes](#)
- [Foundation Documentation](#)

[Back to top](#)

## NSDrawer

`NSDrawer` is a user interface element that attaches to a window, sliding in/out as needed to display other views and controls.

- A crash could occur when using an `NSDrawer` with a text view and the window was closed without being saved. This has been corrected (r. 2696029).
- `NSDrawer` had a 6 pixel by 30 pixel offset. The framework was modified so this no longer occurs. Developers will have to re-link to the new framework to take advantage of the change (r. 2671249).

References:

- [NSDrawer Documentation](#)

[Back to top](#)

## NSFileManager

`NSFileManager` enables you to perform many generic file-system operations.

- `-fileAttributesAtPath:traverseLink:` used to return nil when passed a path of "/dev". It now properly returns an attributes dictionary (r. 2731166).

References:

- [NSFileManager Documentation](#)

[Back to top](#)

### NSFileWrapper

An NSFileWrapper holds a file's contents in dynamic memory. In this role it enables a document object to embed a file, treating it as a unit of data that can be displayed as an image (and possibly edited in place), saved to disk, or transmitted to another application. It can also store an icon for representing the file in a document or in a dragging operation.

- NSFileWrapper now creates compressed .tiff files. In Mac OS X 10.0.x, .tiff files created by NSFileWrapper were not compressed (r. 2722101).

References:

- [NSFileWrapper Documentation](#)

[Back to top](#)

### NSOpenPanel

NSOpenPanel provides the Open panel for the Cocoa user interface. Applications use the Open panel as a convenient way to query the user for the name of a file to open. The Open panel can only be run modally.

- Tagged controls in NSOpenPanel and NSSavePanel now work as expected (r. 2471276).

References:

- [NSOpenPanel Documentation](#)

[Back to top](#)

### NSSavePanel

An NSSavePanel object manages a panel that allows users to specify the directory and name under which a file is saved. It supports browsing of the file system and it accommodates custom accessory views.

- NSSavePanel would sometimes not point to the proper default directory after previous uses. This has been fixed (r. 2739034).
- NSDocument-based applications are now able to save documents correctly even if a parent directory has been renamed or moved (r. 2529722).
- Hidden file name extension support has been added to NSSavePanel (r. 2575042).

References:

- [NSSavePanel Documentation](#)

[Back to top](#)

### NSScrollView

An NSScrollView allows the user to scroll a document view that's too large to display in its entirety.

- Ganged scroll bars have been implemented for NSScrollView (r. 2541625).

References:

- [NSView Documentation](#)

[Back to top](#)

### NSSplitView

An NSSplitView object stacks several sub views within one view so that the user can change their relative sizes.

- A rare case where NSSplitViews would not redraw correctly has been fixed (r. 2505922).

References:

- [NSSplitView Documentation](#)

[Back to top](#)

## NSString

The NSString class declares the programmatic interface for an object that manages immutable strings.

- When doing NSString -> NSData conversion using NSASCIIEncoding, line separators are now converted properly (r. 2706916).
- -[NSScanner isAtEnd] auto-released memory as it went, which for long string processing, could eat up a lot of memory. This operation has been streamlined so it does not use as much memory when processing long strings (r. 2661434).
- Converting to ASCII (using NSASCIIStringEncoding) with lossy conversion now produces more accurate results (r. 2641320).

References:

- [NSString Documentation](#)

[Back to top](#)

## NSTableView

An NSTableView object displays record-oriented data in a table, and allows the user to edit values and resize and rearrange columns.

- NSTableViews can now be resized in drawers (r. 2501737, 2704541).
- NSTableView now allows dragging from windows that are not front-most (r. 2686096).
- Sometimes when dragging data from a table view, developers might have seen text with partial backgrounds drawn behind them (in the drag image). This has been fixed. The default drag image is now always drawn as the selected cells with no background (r. 2654201).

References:

- [NSTableView Documentation](#)

[Back to top](#)

## NSTabView

NSTabView is a user interface element that permits the use of multiple tabs in a window to cycle through different views.

- In some scenarios, NSTabView will crash when the user switches between tabs. Typically you might see this if the views of your items contained contents that changed each time the user switched between tabs. If your tab items contain an unchanging set of views you will not see this problem. The crasher has been fixed (r. 2733195).
- Bugs with keyboard loops in NSTabViews have been fixed, and they should work properly now (r. 2731276, see the AppKit release notes for details).
- In NSTabView.h, -controlSize: has been defined to return a NSControlSize value rather than a NSControlTint value (r. 2719772).

References:

- [NSTabView Documentation](#)

[Back to top](#)

## NSTextField

An NSTextField is a kind of NSControl that displays text that the user can select or edit, and which sends its action message to its target when the user presses the Return key while editing.

- Text width is now updated correctly when an NSTextField, with wrapping turned on, is resized (r. 2623500).
- NSSecureTextFieldCell no longer assumes the view it is given is an NSSecureTextField (r. 2524116).

References:

- [NSTextField Documentation](#)

[Back to top](#)

## NSTextStorage

NSTextStorage is a semi-concrete subclass of NSMutableAttributedString that manages a set of client NSLayoutManager, notifying them of any changes to its characters or attributes so that they can re-lay and redisplay the text as needed. NSTextStorage defines the fundamental storage mechanism of the Application Kit's extended text-handling system.

- NSTextStorage now accesses read-only documents appropriately (r. 2700735).

References:

- [NSTextStorage Documentation](#)

[Back to top](#)

## NSTextView

NSTextView is the front-end component of the Application Kit's text system. It displays and manipulates text laid out in an area defined by an NSTextContainer, and adds many features to those defined by its super class, NSText.

- Dragging the last line up in an NSTextView could break auto-scrolling. This has been fixed (r. 2727187).
- A problem where NSTextView in a drawer could not be set to the first responder more than once has been corrected (r. 2725609).
- For NSTextViews, `-shouldChangeTextInRange:replacementString:` now checks to see if `-isEditable` is YES (r. 2704908).
- NSTextView now draws its selection highlighting even if it is not the first responder (r.2696367).

References:

- [NSTextView Documentation](#)

[Back to top](#)

## NSToolbar

NSToolbar and NSToolBarItem provide the mechanism for a titled window to display a tool bar area (similar to the tool bar areas displayed at the top part of the Finder's windows).

- Adding a default set to a tool bar (which removes items not in the default set) didn't call `didRemoveItem` as it should have. We now properly generate a `didRemoveItem` notification (r. 2723008).

References:

- [NSToolbar Documentation](#)

[Back to top](#)

## NSToolTip

NSToolTip allows applications to display context sensitive help for user interface elements on the screen.

- Attempts to remove tool tips from a view not installed in a window would crash. This scenario is seen most often when using tab views. Consider, for instance a tab view with views A and B, each containing tool tips. If view B is being displayed, and you remove view A's tool tips your app will eventually crash. This has been fixed. (r. 2736943).
- Memory leaks in NSTooltip have been fixed (rr. 2714923, 2713818).
- NSTooltip has been changed so that if you return `nil` from `-view:stringForToolTip:point:userData:` no tool tip will be displayed (r. 2712566).

[Back to top](#)

## NSView

NSView is an abstract class that defines the basic drawing, event-handling, and printing architecture of an application.

- An unmatched `unlockFocus` message now raises an exception instead of simply logging a warning message (r. 2660151).

References:

- [NSView Documentation](#)

[Back to top](#)

## NSWindow

An NSWindow manages an on-screen window, coordinating the display and event handling for its NSViews.

- The `representedFileName` shown in the window title and window menus now hides the extension if the 'hide extension' attribute is set for the file (r. 2718156).
- A problem where `[NSWindow flushWindow]` was not resulting in a context flush for a custom titled window with no shadow has been corrected (r. 2702733).
- A problem where `[NSWindow windowWillClose]` was being called twice has been corrected (r. 2645726).
- `-[NSWindow setTitle:]` now copies the string you pass in instead of retaining it (r. 2745031).
- A problem where a crash could occur in `[NSWindow initWithWindowRef:]` has been corrected (r. 2743856).
- A crash related to choosing a new responder while closing a window with a drawer has been fixed (r. 2686168).
- A problem where resizing a window after calling `setAspectRatio:` would cause the window to shrink to (0, 0) size has been fixed (r. 2565155).

References:

- [NSWindow Documentation](#)

[Back to top](#)

## Printing (AppKit)

This section talks about changes in the printing facilities available in AppKit.

- A problem where the print progress panel would appear in during some operations where its presentation was not appropriate has been corrected (r. 2586862).
- Fixed problem where drawing an `NSImage` from a PICT file scaled down caused horizontal white lines to appear (r. 2764934).
- Fixed a bug where drawing a picture using `QDDrawPictToCGContext` could crash if the PICT contained any ColorSync profile identifiers (r. 2763840).
- Fixed bug where Print Center complains when you try to send it a `quit` Apple event (r. 2742663).
- The `NSPrintInfo` attributes including `NSPrintJobDisposition` (the `NSPrintSaveJob` value), `NSPrintPagesPerSheet`, `NSPrintReversePageOrder`, and `NSPrintSavePath` are now used by the printing system (r. 2574589).

References:

- [NSPrintInfo Documentation](#)
- [NSPrintOperation Documentation](#)
- [NSPrintPanel Documentation](#)
- [NSPrinter Documentation](#)

[Back to top](#)

## Keyboard UI (Cocoa)

The Cocoa Keyboard UI facility allows users to use `tab`, `shift-tab`, and various control keys (user settable; see Preferences) to navigate between user interface elements.

- New for 10.1.

References:

- Keyboard UI section in the [AppKit Release Notes](#)

[Back to top](#)

---

## CoreFoundation

Core Foundation is a set of APIs used as a common data and service abstraction layer for other high level software facilities in Mac OS X.

- Two new `CFTypes`, `CFReadStream` and `CFWriteStream`, have been added to CoreFoundation. These types allow you

to access various sources (like files or sockets) in an incremental fashion; see the CoreFoundation release notes for full details.

- Memory leaks in the CFMessagePort APIs have been corrected (r. 2447596).

References:

- [Core Foundation Framework Release Notes](#)

[Back to top](#)

## CFNetwork

CFNetwork provides facilities for creating and processing HTTP requests; it is a new API in Mac OS X 10.1, and follows the API conventions of CoreFoundation. It is accessed from the CoreServices framework; link against CoreServices to use CFNetwork.

[Back to top](#)

## CFString

String Services is an API that provides a suite of efficient string-manipulation and string-conversion routines.

- A problem where the CFStringCompareWithOptions routine did not respect the kCFCompareCaseInsensitive when it was provided has been corrected. Now, strings are converted to lowercase before comparisons when this flag is provided (r. 2740030).
- The CFStringUppercase API could overwrite part of the input string with other characters because it assumed that each character mapped to a single character, despite encoding - CFStringUppercase now checks the length of the string and behaves correctly with regards to unicode multi-character symbols (r. 2725743).
- The CFStringConvertEncodingToIANACharSetName API is inconsistent in the case representation of the text it returns - The API now uppercases IANA names before passing them back (r. 2722630).
- A problem where the CFSocketConnectToAddress routine would always fail, if a non-zero timeout was specified, has been corrected. (r. 2703634).
- Performance of conversion between text encodings has been improved in many cases (r. 2691725).
- A problem where the routine CFStringGetMaximumSizeForEncoding routine was failing for all Chinese encodings including kCFStringEncodingEUC\_CN, causing some Unicode text searches to fail, has been corrected (r. 2689640).
- The CFString text conversion APIs were not working correctly with the kTextEncodingMacKeyboardGlyphs encoding. Now they do (r. 2660343).

References:

- [Core Foundation String Services Documentation](#)

[Back to top](#)

## CFURL

URL Services is an API that provides facilities for creating, parsing, and dereferencing URL strings.

- A problem preventing the routine CFURLCreateDataAndPropertiesFromResource from working correctly with http: scheme URLs has been corrected (r. 2730143).
- A problem where CFURLCreateWithFileSystemPathRelativeToBase always sets the URL scheme to file: has been corrected (r. 2723055).
- A problem where CFURLCopyFileSystemPath was not properly converting between ':' and '/' characters has been corrected (r. 2695891).
- A problem where CFEqual would abort unexpectedly, if one of the URLs being compared had a NULL scheme and the other did not, has been corrected (r. 2654386).

References:

- [Core Foundation URL Services Documentation](#)

[Back to top](#)

The Dock provides a convenient graphical interface for launching applications and organizing frequently used items.

- New APIs, `Get/SetApplicationDockTileMenu`, have been added in `MacApplication.h` that provide a way for applications to customize the menu showing in the dock when their dock tile is clicked (r. 2670233).

[Back to top](#)

---

## File Systems

This section discusses changes and improvements in the file systems installed with Mac OS X 10.1.

### AFP Server

Implements an AFP (AppleTalk Filing Protocol) over TCP server that allows AppleShare clients to connect and access files.

- Fixed support for AFP short (8.3) file names, which improves compatibility with third party Windows AFP clients (r. 2738157).
- The AFP server now cleans up its SLP registration when the geographic location changes (r. 2699081).
- The AFP server now updates its NBP registration when the computer name changes (r. 2712013).
- A problem where files deleted from inside of a folder that was being deleted would be listed in the parent folder for a short time after the folder was deleted has been corrected (r. 2717288).
- Fixed a bug that would cause the "execute" permission bit to be dropped from applications copied from the server (r. 2750658).
- Fixed a problem that prevented `PBExchangeFiles` working properly on the server (r. 2762796).
- Worked around a problem which caused desktop database calls to AFP volumes sourced by a server on the same machine to deadlock (r. 2769634).

[Back to top](#)

### AppleShare Client

AppleShare Client file sharing services for computers networked with AFP file servers.

- Fixes were made to eliminate crashes, enhance performance, handle aliases correctly, not truncate files, correctly handle multi-byte characters, handle unexpected disconnected volumes more robustly, and fix Classic problems (r. 2720633, 2727960, 2731669, 2740992, 2752604, 2753767, 2754011, 2758652, 2767707, 2703790).
- Fixed an index calculation problem inside of `getdirenties` over AppleShare (r. 2731219).
- A compatibility problem between Mac OS 9 and Mac OS X aliases in the `AFPXVolMountInfo` copied into alias records referring to remote volumes has been corrected (r. 2624962).
- A problem where attempting to enumerate a folder with 100+ items could result in a server being unexpectedly unmounted has been corrected by setting the upper limit for the reply buffersize in the `FPEnumerate` packet. Now, for AFP 3.0 servers, the upper limit is not allowed to exceed the value `0x7FFF`, and for AFP 2.x the reply buffer size limit is set to `8*578` bytes (r. 2739142).
- Folder action scripts running under Mac OS X that watch for a change to an AFP server volume, such as the deletion or creation of a new file, now work as expected (r. 2733329).
- Fixed several problems that could cause kernel panics that could occur when connected to an AFP 3.0 Server (r. 2731215).
- A problem where a copy of an application would appear to be "in use" after being copied to a server, launched, and quit (r. 2728358).
- The routine `PBHGetFInfoSync` was returning different modification dates on the same file in Classic and Mac OS X. The time values returned have been synchronized (r. 2727044).
- To facilitate the use of custom authenticators to talk with AppleShare servers, a new directory, `/Library/Filesystems/AppleShare/Authentication/`, has been created for placement of custom authentication plugins (r. 2723147).
- A problem where the `PBDTGetCommentSync` would fail to get the finder comment for an AppleShare server volume running under Mac OS 9 has been corrected (r. 2715707).
- Calling `PBVolumeMount` again to mount a volume that has already been mounted by a different user will now return an error unless the `volMountSeparateSession` flag is provided (r. 2712033, 2712025).
- File and Folder permissions and group access rights are now preserved when copying files and folders to and from AppleShare servers. Specifically, the group access rights were not being set properly if the group is unknown at the destination system. Now, the "other" access is mapped to "group" access and "group" access is left as "unknown" access (r. 2709825).
- If you are debugging a remote system that establishes a client connection with a server, `afpfs` reported hundred of lines

of status information to the serial port. Afpfs no longer reports this status information to the serial port debug connection (r. 2702948).

- Fixed a problem with the AppleShare X Client not working with Netware servers using ShareWayIP. The server would be shown as mounted under Terminal, but not with the Finder (r. 2702384).
- Fixed a problem with using an AppleShare X Client to connect to a Windows NT server using ShareWay IP where the volume would be mounted, but none of the contents of the volume were displayed (rr. 2702382, 2702379, 2676134).
- Problems where AFP could hang or kernel panic the system in the following circumstances have been corrected: connecting to some third party servers (r. 2702009), enumerating items on a disk containing numerous files (r. 2704236), copying a file and relaunching the Finder simultaneously (r. 2672905), and during some file copy operations from third party Linux servers (r. 2666595).
- The AppleShare X Client now supports connections to AFP Servers over AppleTalk (r. 2691895).
- Improved the performance of file copies (r. 2689487).
- A problem where the file cache was not being updated correctly when a number of cached and uncached read and write calls were intermixed has been corrected (r. 2687263).
- The AppleShare X Client now supports Windows 2000 AFP Servers (r. 2682327).
- The performance in accessing a server folder which has a large number of files, has been improved (r. 2669253).

[Back to top](#)

## HFS

The Hierarchical File System.

- HFS Dates (modification and creation) were not being calculated correctly in Mac OS X 10.0.x. These dates are now calculated correctly (r. 2731813).
- Performance of the `FSSetForkSize` has been improved for HFS formatted disks (r. 2658517).

References:

- [File Manager Documentation](#)

[Back to top](#)

## HFS+

Mac OS Enhanced volume format, which supports file names of up to 255 Unicode characters, file sizes of up to  $2^{63}$  bytes, and more efficient use of space on large volumes.

- A problem where permissions could be set incorrectly has been fixed (r. 2717619).
- The HFS+ file system now correctly handles names with embedded NULL characters (r. 2690395).
- A problem where `PBSetCatInfoSync` could set the modification date to the current time (rather than using the value provided in the parameter block) has been corrected (r. 2675668).
- VFS file systems are now mountable and display in the desktop. This change included setting and respecting the auto-mount, `MNT_AUTOMOUNTED`, bit as well as the Finder respecting this bit (r. 2689584).
- The first time an HFS+ volume is mounted under Mac OS X, the permissions of the root directory on the volume are set to "unknown" rather than "root/wheel" as they were set in Mac OS X 10.0.x. This allows users to delete files from the root directory on the volume which may have been installed by some other software (r. 2665905).

References:

- [File Manager Documentation](#)

[Back to top](#)

## MSDOS

Volume format used by MS-DOS and Microsoft Windows.

- A memory access exception that could occur with 250MB MSDOS formatted USB Zip drives has been corrected (r. 2678871).
- Problems with mounting and unmounting some MSDOS formatted volumes with unusual volume names have been corrected (r. 2699995).

[Back to top](#)

## NFS

Network Filesystem, originally developed by Sun Microsystems, used in UNIX environments for sharing volumes over TCP/IP networks.

- A number of kernel panics that could occur as a result of NFS connections have been corrected (rr. 2301130, 2686121, 2666793).
- NFS file systems mounted via automount time-to-live causes NFS file operations to fail or hang. Fixed a deadlock in the code that checks for expired AFP mounts. Created a new version of automount (r.2715286).
- A problem preventing some volumes from being unmounted because of a "Device Busy" error has been corrected. This problem was only apparent in Mac OS X 10.0.3 (r. 2715295).
- NFS URLs are now registered with NSL using the numeric IP address instead of the hostname so they will work on isolated networks that do not have access to a DNS (rr. 2674380, 2739756).

[Back to top](#)

## ISO 9660

ISO-9660 is a file system plug-in that allows access to ISO-9660 format volumes (possibly with Joliet extensions). This format is commonly used for multi-platform CDs.

- Improved compatibility with certain Kodak PhotoCDs and Kodak PictureCDs (r. 2649047, 2685812).
- A Joliet CD will now display the Joliet volume name rather than just the ISO name (r. 2690991).
- Fixed a bug that prevented files with composed characters from being copied off a Joliet CD (r. 2713915).

[Back to top](#)

## SMB

Server Message Block, Microsoft's file sharing protocol for Windows.

- Support for SMB is new for 10.1 (r. 2712168).

[Back to top](#)

## UDF

UDF is a file system plug-in that allows access to Universal Disk Format volumes. This format is used for consumer DVD media and can also be used as a multi-platform disk format.

- Fixed a problem with Japanese file names on a UDF volume (r. 2751107).
- The default file type and creator for files without Mac-specific information is now '????'/'????'. This allows higher-level software to do the right thing (r. 2714426).
- UDF now correctly supports files larger than 4GB. Previously, UDF disks with a file > 4GB in size would result in missing files in the directory listing (r. 2723679).
- Previous limit of 31 characters for UDF volume names has been fixed (r.2714475).
- A problem where resolving a parent `dirID` to its name was not working correctly has been fixed. In some cases, this problem prevented aliases from being resolved correctly (r. 2650197).
- Added support for reading UDF media that has a Sparing Table on it (r. 2671221).

[Back to top](#)

## WebDAV

WebDAV stands for "Web-based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.

- Problems including memory and vnode leaks seriously affecting heavy WebDAV usage have been corrected (r. 2724670).
- A problem where the WebDAV client did not recognize files or directories having the same name as their parent directory has been corrected (r. 2698397).
- Attempts to perform write operations on WebDAV read only volumes now return EROFS errors rather than EPERM errors as they did in Mac OS X 10.0.x (r.2637743).
- Previously when sending PROPFIND requests the type was not identified in the request as XML. Now it is (r. 2578315).

[Back to top](#)

---

## Graphics

Routines for drawing and displaying graphical information on the screen are discussed in this section.

### ColorSync

ColorSync provides system-level color management that enables publishing software to achieve repeatable and consistent color on-screen, in print, and for electronic delivery.

- `CMGetProfileDescriptions` now returns the correct results (r. 2725835).
- Calling `CMSetDefaultProfileBySpace` to set a version 4.0 profile now works as expected (r. 2724082).
- ColorSync Utility can now change the current profile for a device (r. 2709306).
- ColorSync Utility now supports ICC 4.0 profiles (r. 2690776).
- ColorSync now supports the ICC 4.0 profile specification (r. 2690772).
- A problem where calling `CMSetDeviceDefaultProfileID` could cause problems with the `CMIterateColorDevices` routine (r. 2684661).
- A problem where a dialog background would be drawn black after selecting certain ICC profiles containing Lab cluts has been corrected (r. 2666897).
- A problem where in some cases `CMGetDefaultProfileByUse` would not return the correct profile has been corrected (r. 2633168).
- ColorSync now complies with ICC clarification regarding 16bit Lab data (r. 2615101).
- The `CMDeviceInfo` structure passed to the callback for `CMIterateDeviceProfiles` now has user and host set correctly (r. 2590873).

References:

- [ColorSync Manager Documentation](#)

[Back to top](#)

### Draw Sprocket

DrawSprocket provides drawing services coordinated with the display manager that are useful in game development.

- Problems with hot key handling have been corrected in the `DSProcessEvent` routine (r. 2730907).

References:

- [Game Sprockets Documentation](#)

[Back to top](#)

### ImageCapture

The Image Capture architecture provides a common API for discovering digital imaging devices and transferring images from them.

- "Build web page" script has scripting errors. Fixed a bug in this script which would cause it to fail (2738005).
- Disposing a property would cause the Image Capture extension to quit. Error -9906 now returned when trying to dispose a device property (2733109).
- Wrong profile embedded into images. The profile selected in the Image Capture application preferences is now correctly embedded into downloaded images (2731774).
- Project Builder error, "ICAApplication.h:428 rotationAngle has non-optimal alignment" when building Image Capture applications. Fixed incorrect data type in ICAApplication.h (2731547).
- Sending the message "del1" does not work for mass storage cameras. Added support for deleting images from mass storage camera devices (2730929).
- Fixed a bug where no error is being returned on `ICDownloadFile` if the object is nil (2724105).
- Fixed a bug where calling the Image Capture APIs in a thread other than the main thread would crash in Apple Event processing (2722785).
- Fixed a bug where the Image Capture AppleScript "Build Web Page" would error if the user set the scripts settings to not show image names (2647225).
- Can't target Image Capture Extension for scripting. It has now been moved to `/System/Library/ScriptingAdditions`, which is the designated place for scriptable pieces of components (2613463).

[Back to top](#)

## OpenGL / OpenGL Carbon / AGL

OpenGL is a vendor-neutral, multi-platform graphics standard with broad industry support. OpenGL APIs allow you to develop portable, interactive 2D and 3D graphics applications.

- Improvements to display list execution when using `GL_LINE_STRIP` (r. 2736531).
- A problem where calling `glEnd` could crash malloc debug has been fixed (rr. 2631486, 2698112).
- A problem rendering borders on 2D textures has been fixed (r. 2663147).
- `glTexSubImage2D` and `glPixelStorei` with `GL_UNPACK_ROW_LENGTH` now work when using `GL_UNSIGNED_INT_8_8_8_8` pixel format with `kCGLPFANoRecovery` on (r. 2663528).
- Various OpenGL improvements under Classic environment have been implemented (r. 2665539).
- Calls to `glDrawBuffer(GL_NONE)` has been fixed for double-buffered contexts (r. 2679450).
- A problem using full-screen contexts on multi-monitor macs has been fixed (r. 2688559).
- Support for shared texture palette extension has been added (r. 2690757).
- Some performance optimizations to Pixel Transfer, Immediate Mode, State Management, and `glDrawPixels` have been implemented (rr. 2690754, 2702570).
- A problem causing Carbon applications to display the alert "<AGL.vlib missing>" has been fixed (r. 2709433).
- Calls to `glDepthMask(GL_FALSE)` now work as expected (r. 2719700).
- A problem with surface updates on windows with multiple GL/AGL contexts has been corrected (rr. 2571842, 2742836).
- A problem where calls to `aglDescribePixelFormat` could return zero for supported renderers has been corrected (r. 2709183).
- Improved performance and interoperability with the NVIDIA GeForce video card (rr. 2688069, 2746882, 2614345, 2646245, 2649681, 2669524, 2672996, 2673055, 2710243, 2710896, 2715900, 2715978, 2716708).

References:

- [OpenGL Documentation](#)

[Back to top](#)

## Printing

This section discusses new features, bug fixes, and new APIs in the Mac OS X 10.1 printing services.

- **New features**
- Auto discovery of USB printers has been added (r. 2514274).
- Print Center icon in the Dock is now badged to show job status (number of sheets to be printed) and errors, and its contextual menu provides access to the printer list (r. 2559380).
- Save as PostScript file added as an output option (r. 2696351).
- A problem where print dialog overrides were not working correctly under certain conditions has been corrected (r. 2649798).
- A problem where switching printers in the Page Setup dialog would cause the document's paper size to change has been corrected (r. 2630468).
- A problem where switching from US English to a European language did not change default paper size from US Letter to A4 has been corrected (r. 2707916).
- Problems with USB bus arbitration between X and Classic during printing have been corrected (r. 2634123).
- PAP EOF flag was not being set for last data packet of a query job in PAP connections generated by the Print Center. This has been corrected (r. 2642999).
- Support for Quickdraw CopyBits transparent mode has been added (r. 2622042).
- Calls CopyBits to print 1-bit images using the `crcCopy` transfer mode were not being rendered correctly when the foreground color wasn't black and/or the background color wasn't white. Now they are (r. 2622009).
- Calls to CopyBits using the `srcBic` transfer mode to print images with a bit depth greater than one were not being imaged correctly. Now they are (r. 2699025).
- Support for MM (Multiple Master) fonts has been added (r. 2571329).
- Support for fonts with derived styles and fractional widths has been added (r. 2700792).
- Support for 8-bit PS generation has been added (r. 2563000).

- Support TBCP for PS over USB has been added (r. 2564057).
- The search for PPD files has been extended to include the Classic System folder, after searching the Mac OS X System and Local domains (r. 2611301).
- Installation of localized PPD files is now supported by organizing PPD directory tree like a Mac OS X bundle (r. 2621629).
- PS queries now include the %!PS-Adobe-3.0 Query (r. 2642964).
- Problems with the LW8 compatibility path failed that could occur when aType 1 font could not be found in /System/Library/Fonts/ or the job contained only OTF fontshave been corrected (rr. 2657537, 2658079).
- HP, Lexmark and Xerox PPD files are now bundled with Mac OS X (r. 2671153).
- PostScript jobs did not emit PPD code for \*DeviceAdjustMatrix. Now the do (r. 2706518).
- A new API, `PMSessionCreatePageFormatList`, to support finder tuned controlover paper sizes has been added (r. 2568970).
- New APIs that allow callers to set the system to write finished PostScript files to Disk have been added. These APIs include `PMSessionSetDestination`, `PMSessionGetDestinationType`, `PMSessionGetDestinationFormat`, and `PMSessionGetDestinationLocation` (r. 2696352).
- New APIs, `PMSessionCreatePrinterList` and `PMSessionSetCurrentPrinter`, have been added for accessing the system's print queues (r. 2651979).
- `PMSessionBeginDocument` now returns `kPMNoDefaultPrinter` when there are no print queues (r. 2703817).

References:

- [Printing Documentation](#)

[Back to top](#)

## Quartz 2D

Quartz 2D is a feature-rich, two-dimensional drawing engine that is accessible from all Mac OS X applicationenvironments outside of the kernel. The Quartz 2D application programming interface (API) is easy to use and gives youaccess to powerful features such as Bezier curves, path-based drawing, transparency, and advanced color management.

- A problem where `CGContextClip` would not clip properly during somedrawing operations involving mixing of alpha masks and shapes whenanti-aliasing was turned off has been corrected (r. 2726562).
- The `CGPostKeyboardEvent` has been updated so it no longer includesmodifier keys that are being held down when it is called. Other issues related tospecifying modifier keys with `CGPostKeyboardEvent` have been corrected as well (rr. 2717989, 2706275).
- A new API has been added, `CGColorSpaceCreateWithPlatformColorSpace`, that allows `CGColorSpaceRefs` to be created from `ColorSync` profiles (r. 2715298).
- `CGPostKeyboardEvent` does not work as expected. Changes made to `CoreGraphics` to fix this issue (r. 2706275).
- A problem where invalid memory accesses could occur whenrendering a `CGImage` in cases where its row stride (`bytesPerRow` parameter of `CGImageCreate`) is larger than (`bitsPerPixel * width / 8`) has been corrected (r. 2696962).
- APIs for using Patterns for filling areas have been added (r. 2692348).
- A memory leak that could occur in Carbon applications when the bit depth ofthe display is changed has been corrected (r. 2667929).
- A problem that could occur when drawing zero length lines when renderinga PDF file has been corrected (r. 2662308).
- A new API has been added, `CGWindowServerCFMachPort`, that returns thewindow server's `CFMachPortRef`. This value can be used by Carbon orCocoa applications to obtain port-death notifications so they can establishwhen the window server is closed (r. 2643122).
- The performance of the routines `CGDisplayMoveCursorToPoint`,and `CGWarpMouseCursorPosition` has been improved (r. 2545018).
- A problem where setting text rendering to `CGFilledClippedTextRenderingMode` or `CGFilledStrokedClippedTextRenderingMode` and then calling `CGShowTextCGShowTextAtPoint` could resultin an invalid text clipping region has been corrected (r. 2386257).

References:

- [Quartz 2D Documentation](#)

[Back to top](#)

## QuickTime

QuickTime provides various multimedia services for the Mac OS, including the ability to display movies and facilities forthe translation and display of various audio and visual data file formats.

- Mac OS X 10.1 ships with QuickTime 5.0.2. See the QuickTime release notes for details.

References:

- [QuickTime Documentation](#)

[Back to top](#)

---

## Hardware/Devices

This section discusses changes and new features for device driver writers provided in Mac OS X 10.1.

### Disk Arbitration

Disk Arbitration manages disk insertion and ejection. Amongst other things, it probes newly inserted disks to determine their file system format, checks these disks for corruption, and mounts the file system on the disk.

- Disk Arbitration will now mount removable media by default if the setting in `/etc/hostconfig` is lost (r. 2694226).
- Disk Arbitration will no longer delete mount points it didn't create (r. 2710178).
- You can now press the keyboard Eject key to inject an open CD/DVD Tray (r. 2739698).
- Made numerous improvements in how disks are mounted and unmounted (rr. 2741930, 2741621, 2741570, 2740736, 2740326, 2738856, 2732894, 2732806, 2715924, 2731111, 2727289, 2725495, 2722929, 2722826, 2718662, 2717648, 2716305, 2714577, 2713027, 2710467, 2694976, 2694772, 2694668, 2694622, 2691584, 2691553, 2691551, 2690407, 2682158, 2681271, 2679446, 2653158, 2647679, 2586029, 2583014, 2559644, 2475229).
- You can now send a SIGHP to the Disk Arbitration server (autodiskmount) to force it to refresh its internal state (r. 2426274).
- Disk Arbitration now supports an option to always perform file system checks (fsck) on read-only media (r. 2685016, 2503133).
- Disk Arbitration now identifies CD media in a DVD drive as a CD, not a DVD (r. 2507759).
- A problem preventing users from renaming a hard disk to have more than 10 Japanese characters has been corrected (r. 2754272).

[Back to top](#)

### IOKit

Apple's object-oriented I/O development model. The I/O Kit provides a framework for simplified driver development, supporting many families of devices.

- `kIOMessageServiceIsAttemptingOpen` general interest notifications now include the options argument passed to `IOService::open` (r. 2733924).
- The performance of the `IORegistryEntryCreateCFProperty` routine has been improved (r. 2670274).
- Moved the implementation of `readBytes` and `writeBytes` from `IOGeneralMemoryDescriptor` to `IOMemoryDescriptor`. As a result, subclasses of `IOMemoryDescriptor` no longer need to implement their own versions of `readBytes` and `writeBytes` (r. 2738315).
- The `IORegistryEntry::matchPathLocation` call now properly handles 64 bit locations (r. 2718611).
- The PCI family has been changed to be loadable. Drivers now need to explicitly declare their dependency on `IOPCIFamily` per IOKit documentation and release notes (r. 2699334).
- Changes were made to make some kernel level drivers loadable thus lowering the wired memory footprint (rr. 2370069, 2370075, 2370078...).
- A problem where `IOFramebuffer::deliverFramebufferNotification` always returned `kIOReturnSuccess`, even in cases where one of the notification handlers returned an error, has been corrected (r. 2602514).
- Problems with recognizing video cards supporting only 24 bit color have been corrected (r. 2665190).
- Declarations for `IORegistryEntryGetLocation` and `IOObjectRetain` have been added to `IOKitLib.h` (r. 2670078).
- Additional boot time debug info is now provided to `IOLog` when the i/o kit debug flag is enabled (r. 2686931).
- IOKit drivers are now able to register with the root power domain for power notifications such as shut downs and restarts (r. 2663574).

[Back to top](#)

## KEXT Management

Services for loading and unloading kernel extensions.

- The kalloc map was increased from 8MB to 16MB. The limit had been reached by large kexts such as debug versions with symbolic information (r. 2699907).
- The performance of kernel extensions loading at boot time has been improved (rr. 2702018, 2691879).
- Kernel extensions will no longer load if they fail the version check. On Mac OS X 10.0.x a warning was issued but the kext would load anyway (r. 2605536).
- Most KEXTs that depend on BSD internal data structures need to be rebuilt. Also, for KEXTs to be loaded they must declare their dependency on com.apple.kernel.bsd to be 1.1; if the dependency is not set to this version they will not be loaded. For more information refer to [BSD Kernel](#) section.

[Back to top](#)

## Mass Storage

Drivers for storage devices such as hard disks and CD-RW drives.

- Problems preventing the data volume from being mounted on some multisession CDs have been corrected (r. 2678448).
- Problems experienced by some developers with the `readCD` have been corrected (rr. 2730219, 2748293).
- Fixed problem where only the first partition of certain multi-partition CD-R media would mount (r. 2720421).
- Added support for MO drives and USB floppy drives (rr. 2768365, 2771244).
- Added DVD playback support on selected models (r. 2546553).

[Back to top](#)

## PCCard Support

Drivers for PCCard and CardBus card devices.

- Add support for multifunction cardbus cards (rr. 2723252, 2708051).
- A problem where PCCards were not being powered off during sleep has been corrected (r. 2616128).

[Back to top](#)

## SCSI Architecture Model (SAM)

Support for devices conforming to the SCSI Architecture Model. This includes ATAPI, FireWire, and some USB storage devices.

- `IO SCSIProtocolInterface::GetUserClientExclusivityState` and `IO SCSIProtocolInterface::SetUserClientExclusivityState` are now public virtual functions (r. 2768807).
- Added ability to distinguish DVD-RAM support in the "DVD Features" property (r. 2720961).
- Added `kCDFeaturesCDDAStreamAccurateBit` to the `CDFeatures` property so applications can find out if a CD drive doesn't need stitching (r. 2764753).
- Eject key now opens CD tray on drives that don't respond properly to `GET_EVENT_STATUS_NOTIFICATION` commands (r. 2749692).
- `IO SCSIMultimediaCommandsDevice::readMCN` and `IO SCSIMultimediaCommandsDevice::readISRC` now work as documented (r. 2730240).
- A problem where the Vendor Identification string in the IORegistry was set to garbage characters for devices with a vendor identification string of all spaces has been corrected (r. 2726934).
- A new property, `SCSITaskDeviceCategory`, has been added to property `IO SCSIPeripheralDeviceNub`. Also, new definitions for the objects `IOCompactDiscServices` and `IODVDServices` have been added to the interfaces (r. 2724405).
- `IO SCSIPeripheralDeviceNub` objects now have a GUID property (r. 2724404).
- Device Characteristics and Protocol Characteristics dictionaries have been added to the `IOBlockStorageDevices` definitions. This change provides a means to determine how a device is connected and what CD and DVD features the device supports. The new keys are defined in `IOKit/storage/IOStorageDeviceCharacteristics.h` and other related keys are defined in command set-specific headers, such as `IOKit/scsi-commands/IO SCSIMultimediaCommandsDevice.h` (rr. 2706348, 2706360).
- A problem where errors of type `-36 (ioErr)` could be reported during file copies to FireWire hard drives as a result of

- a FireWire bus reset has been corrected (r. 2653050).
- SCSI Application Layer now supports client request timeouts (r. 2648536).
- New CD Get/SetMediaAccessSpeed APIs have been added to IOSCSIMultimediaCommandsDevice (r. 2730472).
- Added SCSI Task User Client. This allows user space code to send SCSI Tasks to SCSI Primary Commands compliant devices with Peripheral Device Types other than 00h, 05h, 07h, and 0Eh. Devices which have authoring capabilities, namely CD-R/W and DVD-R/W devices, can also use the SCSI Task User Client to send commands to a device (r. 2599900).

[Back to top](#)

---

## Java

This section discusses changes and fixes present in the Java implementation provided in Mac OS X 10.1.

- A crashing problem in `java.io.File.getCanonicalPath()` was corrected (r. 2733452).
- A problem where `Locale.getDefault` was always returning the value "en" as the default locale has been corrected. `Locale.getDefault` now returns a meaningful value (r. 2667503).
- A problem where Java applications would fail to launch when machine date was set to before 1/1/1970 has been corrected (r. 2639678).

## AWT

Abstract Windowing Toolkit (AWT) is a cross platform windowing system used in Java applications. Swing is built on top of AWT.

- Java windows did not refresh after hiding, they now do (r. 2746001).
- Many Drag and Drop issues have been fixed (r. 2681079).
- `FileDialog` now allows resizing (r. 2528156).
- `AWTFileDialog` used to ignore `setDirectory()`, now it does not (r. 2735612).
- Click-to-go-here feature has been implemented for AWT scroll bars (r. 2732106).
- Heavyweight and Lightweight components didn't mix in `CardLayout`, they now do (r. 2721431).
- Placing a `Frame` outside the screen area caused it to appear at (0,0). This has been corrected so now it will not appear on screen (r. 2720752).
- Setting a `Frame` or `Dialog`'s `setResizable` false and then calling `setSize()` would cause a crash. This has been corrected so now it has no effect (r. 2720630).
- Previously the `com.apple.macosx.AntiAliasedGraphicsOn` property had no affect. Now it turns on anti-aliased graphics drawing (r. 2719642).
- Previously the `com.apple.macosx.AntiAliasedTextOn` property had no affect, but it now turns on anti-aliased text drawing (r. 2719641).
- Problems where invisible windows could cause crashes under certain circumstances have been corrected (r. 2719588, 2719589).
- A problem where repeatedly showing or moving a `JDialog` could cause a crash has been fixed (r. 2719492).
- A problem where `JDialog` did not disable menus properly has been corrected (r. 2712885).
- The Print progress dialog was not properly re-enabling menus after being dismissed. Now it does (r. 2712865).
- A problem where Window bounds were not being updated after moving a window has been corrected (r. 2709420).
- The `Frame` Peer used to throw an exception when the `Frame` got focus while adding menus, but it no longer does (r. 2709311).
- List navigation using the arrow keys moves the selection to the next item, in Mac OS X 10.0.x it moved the selection 2 cells away (r. 2703456).
- `FileDialog` returned a mangled name for files that have 8-bit characters in their name, now it returns a proper UTF8 name (r. 2699436).
- Under some circumstances Modal Dialogs and `JDialogs` could freeze an application, but they now behave correctly (r. 2698180).
- Invoking the `translate` method on a Graphics object created from a printer context didn't work correctly, but now it does (r. 2697592, 2698221).
- Mac OS X Java Frames that were not supposed to be resizable could still be maximized, but now they can not (r. 2696978).
- `DropTargetDragEvent` method `getLocation()` used to return wrong position (local not global), but now it returns a global position (r. 2696171).

- Disabled menu's could become enabled when the menu is clicked on, now they stay disabled (r. 2696151).
- Cursor shape was not being updated after window deactivate/reactivate. Now it is updated (r. 2689283).
- The JToolBar window did not dock/undock properly (it drew behind other windows), but now it behaves correctly (r. 2669608).
- Disabled menu items with accelerators caused hangs in 10.0. Now they behave correctly (r. 2659749).
- A Java Application could not receive Apple events unless it made a call to `CPSRegisterWithServer(carbon flavor)` before loading AWT lib. In 10.1 this is no longer necessary (r. 2658960).
- The Toolkit Peer's color setting code wasn't set thread safe in Mac OS X 10.0.x. Now it is (r. 2651909).
- `MRJApplicationUtils.registerQuitHandler` would throw a `NullPointerException` if called before AWT was loaded. Now it does not (r. 2651622).
- Java apps sometimes crashed when attempting to access disposed windows. They no longer do (r. 2644062).
- With a default `FileDialog`, the user could not select a `.app` application. Now they can (r. 2614649).
- A hang that could occur on Quit has been corrected (r. 2614602).
- Labels did not change size to reflect updated string length, but they now do (r. 2583125).
- List peer deselect in single-select mode didn't check if the index was valid, but now it does (r. 2539318).
- Passing focus to a window that was being disposed could cause a null pointer exception. Now it will not (r. 2516328).

[Back to top](#)

### Bridge Technology

Java Bridge Technology allows Java classes to subclass Objective-C classes and reference Objective-C classes.

- Objective-C calling Java sometimes would invoke the wrong method. We now do a better job at making sure the signatures match (r. 2608405).
- Since you cannot use the `==` operator to compare `NSSelectors` an `isEqual` method has been added for comparisons (r. 2252809).
- The Foundation Java initialization routines were not being called by the Java/ObjC bridge, so morphing was not being setup properly for any of the morphed objects (`NSDecimalNumber`, for example). Now the bridge explicitly loads dependent libraries (r. 2308011).
- Asking an NS object for its URL would throw an exception if the URL was `NULL`. Rather than throwing an exception, it now returns `NULL` (r. 2665340).

[Back to top](#)

### Embedding

Java Embedding is a framework that allows Java Applets to be embedded inside of Carbon applications.

- A problem where embedded applets without focus could intercept user input events has been fixed (r. 2511098).
- Various reliability improvements for Java embedding have been implemented (rr. 2519898, 2600213, 2638472, 2651426, 2679378, 2689641, 2693347, 2697511, 2697518, 2697524, 2697518, 2708134, 2704924, 2709852, 2711898, 2720805, 2723293, 2743619).
- Support for `showDocument()` and `showStatus()` have been added for use by web browsers (r. 2545471).
- Libraries pertaining to Java embedding now reside in their own Framework (rr. 2703482, 2578808, 2733977).
- A problem where Internet Explorer gave incorrect coordinates for positioning Java applet controls has been fixed (r. 2580608).
- HTTP authenticator support has been added for embedded applets (rr. 2629138, 2693417).
- A problem where Internet Explorer called `applet.stop()` when an applet's host window lost focus, causing flickering, has been fixed (rr. 2670878, 2670962).
- Embedded applets are now clipped properly when resized or moved (r. 2671085).
- Embedding calls can now be made from multiple threads (r. 2674470).
- Applet parameters are now read from Internet Explorer (r. 2674474).
- A problem where boxes intermittently appeared between text typed in applet `TextField`'s has been fixed (r. 2674504).
- A problem where the first `mouseReleased` event in an applet could go to the wrong component has been fixed (r. 2692363).
- A problem where returning to a page with a Java applet could crash Internet Explorer has been fixed (r. 2693344).
- A problem where applets output could overwrite the toolbar in Internet Explorer has been fixed (r. 2701402).

- A problem where applets in Internet Explorer with focus were not recognizing key events has been fixed (r. 2713719).
- A problem where the graphics port could change when making embedding calls from a browser has been fixed (r. 2721668).
- Deadlocking problems when scrolling windows with embedded applets have been fixed (rr. 2734901, 2734904).
- A problem where null applet URLs could cause a crash has been fixed (r. 2741437).

[Back to top](#)

## Graphics

Changes to Java's graphics implementation that are relevant to developers are discussed in this section.

- Various improvements to XOR drawing mode have been implemented (r. 2499676, 2636026, 2643300, 2684230, 2684233, 2699353, 2702077).
- Various improvements in Java, Java2D drawing performance have been implemented (rr. 2539624, 2597968, 2650703, 2658375, 2671016, 2671028, 2671074, 2686338, 2704952, 2713795, 2713875, 2727227).
- Various improvements in GUI component drawing have been implemented (rr. 2627491, 2693950, 2724572).
- Various improvements in Java2D printing have been implemented (rr. 2561648, 2603766).
- A problem preventing TogetherJ from running on Mac OS X has been fixed (r. 2680016).
- A problem that could crash Applet Launcher under certain locales has been fixed (r. 2703835).
- `BufferedImages` with byte `DataBuffers` are now supported (rr. 2669214, 2681035).
- Fixed a bug where `drawRect()`'s line thickness inconsistent (r. 2752733).
- Fixed a bug where XOR mode drawing doesn't work when a `JComponent` is inside a `JScrollPane` (r. 2724599).
- Fixed a bug where images with an alpha channel were not anti-aliased properly (r. 2719498).
- Changed `drawImage` to handle the case with `dst` flipped over the y-axis (r. 2717089).
- Fixed bug where XOR drawing did not render correctly for some color combinations when drawing in thousands-of-colors mode (r. 2657050).

[Back to top](#)

## HotSpot

HotSpot the Java virtual Machine used in Mac OS X.

- The JNI function `AttachCurrentThread` was modified to allow previously created `pthread`s to be attached as daemon threads (r. 2717431).
- A problem where a `SIGTRAP` exception could occur while using JVMPI instruction tracing has been corrected (r. 2716155).
- A problem that could result in an exception being thrown when element 32768 of an array was accessed has been corrected (r. 2699669).
- Spurious `ArrayOutOfBoundsException` and `NullPointerException` exception that could occur for valid objects have been corrected (r. 2681318).
- A problem where repeated JVMPI suspend/resume operations on threads could cause them to get stuck in calls to `wait()` has been corrected (r. 2677945).

[Back to top](#)

## Swing

Platform Independent GUI APIs for Java.

- A problem where Drag & Drop could throw spurious exceptions has been fixed (r. 2690650).
- When dragging from lightweight components, e.g. `JTextField`, the `dragrectangle` used to be positioned outside the window with varying offsets. Now it is positioned correctly (r. 2688201).
- Mouse right-button clicks now work as expected (r. 2669617).
- Large `TooTips` would just appear then disappear, they now stay around as expected (r. 2611120).
- Selecting a `PopupMenu` item in an applet resulted in "VToolkit.postEvent: Unknown ApplicationContext for Deferred". It is now fixed (r. 2557368).
- Java did not allow switching keyboard script with command - space, it now does (rr. 2682687, 2667295, 2699301).

- Swing Applets used to cause `SecurityManager` exceptions accessing internal properties, this no longer occurs (r. 2620186).
- `FileDialog` can't choose an application under Mac OS X 10.0.x, but it now can (r. 2546407).
- `FileDialog` didn't remember where it's been; and it always reset to root of drive. It now remembers the last location used (r. 2693988).
- Problems rendering popup menus on `JComboBox` components when using dual monitors have been fixed (r. 2754051).
- A problem where non-focused `JTextFields` could receive keystrokes has been fixed (r. 2721748).
- A problem where `JComboBox` components could be sized incorrectly has been fixed (r. 2681582).
- Composite `JButton` components now render and function properly when placed inside a `GridBagLayout` (r. 2679273).
- A problem where `JMenuItems` relocated to the Mac OS screen menu bar stop firing `mouseEntered/Exited` events has been fixed (r. 2679250).
- Problems with cut-off event handling for `JWindow` and `JDialog`-based components have been fixed (r. 2677522).
- Fixed a bug where `JTabbedPane`'s tab heights were drawn too small for Web Objects Java clients (r. 2738236).
- Fixed bug in `KeyStroke.getKeyStroke()` and `JMenuItem.setAccelerator` (r. 2737102).
- `JScrollbar`s now work as expected (r. 2733748).
- `setEditable` in `JTextComponent` doesn't work as expected. Fixed problem with selected text (r. 2733214).
- `JMenu` can now hide the menu bar by setting the visible property (r. 2717182).
- `JInternalFrame.isPalette` used to be the way to get palette style frames, now it's deprecated in favor of `JInternalFrame.frameType - normal, palette or optionDialog` (r. 2716333).
- `JToolBar.setMargin()` now works as expected (r. 2709181).
- Fixed bug where a zero-length string cannot be added to a `JComboBox` (r. 2708272).
- A problem where text was being cut off instead of being displayed below the `JProgressBar` in calls to the `setString` method of `JProgressBar` has been corrected (r. 2700364).
- The `UIManager.getLookAndFeel().getID` method now reports a standard ID code (r. 2694545).
- A problem where `JFileChooser` was not passing directories to `FileFilters` has been corrected (r. 2689158).

[Back to top](#)

References:

- [Java Developer Documentation](#)
- [Mac OS X Java GM Release Notes](#)
- [General Java Information Release Notes](#)
- [Java Cocoa APIs Release Notes](#)

[Back to top](#)

---

## Kernel

The complete Mac OS X core operating system environment that includes Mach, BSD, the I/O Kit, file systems, and networking components.

### BSD Kernel

BSD (Berkeley Software Distribution), formerly known as the Berkeley version of UNIX, BSD is now simply called the BSD operating system. The BSD portion of the Mac OS X kernel is based on FreeBSD, a version of BSD.

- Problems with the "%q" format designator in `scanf` have been fixed (r. 1685258).
  - `/dev/random` is now implemented. It uses the Yarrow algorithm and is seeded automatically by the system. `/dev/urandom` is also implemented for compatibility with Linux/BSD (r. 2288835, 2705720).
- NOTE: the man page for `random` that shipped with Mac OS X 10.1 was not updated in time to describe these features.
- The `issetugid` system call was added (r. 2720059).
  - Cases where creating hard links to nonexistent or fifo files would panic the kernel have been corrected to report an error as "operation not supported" (r. 2668770).

#### IMPORTANT:

In order to fix some serious and fundamental problems with `select(2)` on dual processor systems, it is necessary to introduce a binary incompatibility in a few KEXTs for Mac OS X 10.1. Only KEXTs that depend on BSD internal data structures are affected. Most just would need rebuild with the latest Mac OS X 10.1 header files. There is no impact on user level binaries at all.

The changes in the kernel that affect compatibility are:

1. `selinfo` data structure has changed. It now includes `wait_queue` and a flag and all other previous fields are removed. This data structure is part of `struct tty` and `struct sockbuf`.
2. `fileops select` function now takes an extra parameter which is a `void *`. This extra argument is passed to `selrecord()`.
3. Also the `selrecord()` now uses another argument which is a wait queue link to link the wait queue to its subordinate. This argument is passed to the `fileop select` function which in turn passes this to `selrecord`.
4. Callers of `sopoll` from the kernel would need changes, they should refer to `smb source`, which has all the changes for this.
5. Also the kext that declare dependency on `com.apple.kernel.bsd` should now have that dependency to be declared for version 1.1. Otherwise it will not load.
6. Many networking data structures are padded and some macros are converted to functions to avoid breaking compatibility in future. So for this just need to rebuild the KEXT with latest Mac OS X 10.1 headers.

What the developers need to do:

1. Most developers need to rebuild the kernel extensions with Mac OS X 10.1 or greater header files. Also declare dependency on `com.apple.kernel.bsd` to be 1.1.
2. If you have calls to `selrecord()` in your code then please pass the extra argument you get passed in.
3. if you are calling `sopoll()` in your code then refer to `smb project` available thru Darwin to see the changes you would need to do.

[Back to top](#)

#### Mach Kernel

The lowest level of the Mac OS X kernel. Mach provides such basic services and abstractions as threads, tasks, ports, IPC, scheduling, physical and virtual address space management, VM, and timers.

- Several improvements to the virtual memory management logic have been implemented to improve stability under heavy virtual memory usage conditions (r. 2770089, 2770089, 2682033, 2759791, 2668080).
- New APIs have been added to `mach/thread_policy.h` that allow for additional control over task scheduling (r. 2729073, 2720253).
- A number of performance improvements have been implemented that speedup boot time, real-time threads, thread management, cache flushing, and preemption handling (r. 2722814, 2709447, 2693798, 2614992, 2730178).
- A problem where, under certain conditions, the `trunc` function was not returning the correct value have been corrected return incorrect value (r. 2704147).
- The command line tools `uptime` and `w` are now more accurate (r. 2685211).
- Process quantum calculations now use higher resolution time units than before (r. 2418966)

[Back to top](#)

---

## Launch Services

Launch Services provides APIs applications can use for launching other applications.

- A new finder flag, `LSInfo`, has been defined in `Finder.h` that can be used to determine if an item has a hidden file name extension (r. 2575312).
- The `LSGetApplications` routine has been replaced by a new routine, `_LSCopyAllApplicationURLs`, that provides better error checking (r. 2499328).

References:

- [File Name Extension Guidelines Release Notes](#)

[Back to top](#)

---

## LoginWindow

LoginWindow provides login facilities at system startup time and between user login sessions.

- Previously if an application was killed via terminal or some other method you would sometimes get the message from loginwindow indicating that the 'application unexpectedly quit'. This problem was fixed by not reporting the error when the termination of the application is due to a kill (SIGTERM/SIGKILL) (r. 2728153).
- In some cases logout would fail because certain applications were taking too long to shutdown rather than the logout being cancelled. This was fixed by setting logout timeout for each application to 45 seconds (r. 2652139, 2672680).

[Back to top](#)

---

## Networking

This section discusses changes and new features in the networking services provided in Mac OS X.

### AppleTalk

AppleTalk is a built-in network protocol that supports communications over AppleTalk networks.

- A problem where AppleTalk could corrupt data in ATP responses has been corrected (r. 2703163, 2701926).
- A problem where the AppleTalk routine SPRegister could return an error when passed valid parameters has been corrected (r. 2746200).
- Timeout and retry settings for PAP connections have been adjusted for better compatibility with LaserWriter printers (r. 2704409).
- A problem where AppleTalk was demultiplexing DDP packets incorrectly has been corrected (r. 2498601).
- Previously the number of 'interfaces' AppleTalk could accommodate simultaneously was limited to a fixed number (about 30). This was unnecessary and very limiting. This was fixed so now AppleTalk will now work with an arbitrarily large number of interfaces (r. 2655223).

[Back to top](#)

### DHCP and BOOTP

DHCP and BOOTP servers provide automated network address configuration services. This section discusses changes in Mac OS X's facilities for accessing DHCP and BOOTP servers.

- Facilities have been provided that allow developers to ask Mac OS X to request specific DHCP options when requesting a network configuration, and to get the values of those options after the interface has been configured (r. 2684631).
- Improved compatibility with DHCP servers that don't support DHCP INFORM (r. 2767372).

[Back to top](#)

### Directory Services

Directory Services provides an abstraction layer designed to isolate clients of the Directory Services programming interface from the actual implementation of a directory system.

- In searching for standard type user records via `dsGetRecordList`, using the NetInfo plug-in, the native record name types that will be searched against are now first "name" and then next "realname" (r. 2690932).
- If a custom authentication search path is configured using Directory Setup, `lookupd` will now consult that Directory Service search path. LDAP servers configured and added to the custom search path are then available to clients using BSD APIs such as `getpwnam` to access user, group, and mount information (r. 2441856).
- Added new standard types to support integration between `lookupd` and Directory Services:
  - Two new standard types, `kDSLAttrNFSHomeDirectory` which is used for locating a NFS home directory and `kDSLAttrPasswordPlus` which is a password key, were added. Plug-in developers should map `kDSLAttrPasswordPlus` to either the crypt password if available or "\*\*\*\*\*" to indicate that the plug-in requires a direct API call into `dsDoDirNodeAuth` (r. 2695911).
  - New standard types, `kDSLAttrVFSType`, `kDSLAttrVFSPassNo`, `kDSLAttrVFSDumpFreq`,

`kDSIAAttrVFSLinkDir`, `kDSIAAttrChange`, and `kDSIAAttrExpire`, were added to accommodate NFS mounts using Directory Service (r. 2692274).

- Now DirectoryService always checks if the Windows Authentication Manager (tim process) is running and properly enabled before attempting to use it to set or change passwords (r. 2715067).
- The DirectoryService API call `dsGetRecordEntry` now returns the proper errors when invalid buffer formats are detected (r. 2710260).
- Any `NetInfo` nodes that are opened by a client call to `dsOpenDirNode` are now automatically placed in the registered list. This is important for search policy nodes (r. 2703669).
- DirectoryService now supports parsing the results of `dsGetRecordList`, `dsDoAttributeValueSearch`, and `dsDoAttributeValueSearchWithData` entirely within the framework instead of requiring mach message traffic for each parsing API call. This change greatly improves performance when parsing large buffers with many records (r. 2649626).

References:

- [Directory Services Documentation](#)
- [Directory Services Plugin Documentation](#)
- [LDAP Plug-in](#)

[Back to top](#)

## Internet Config

Internet Config provides an API to access and manage Internet-related preferences.

- The `ICLLaunchURL` routine now provides better handling of poorly formed URLs (rr. 2631099, 2691027).
- Internet Config no longer has a default value for the `LanguagePrefs` key. This corrects a problem where the web browser's default language is always set to English (r. 2637703).
- A problem where the `ICLLaunchURL` routine could trigger a GURL Apple event storm when the application sets the preferred helper application does not have a GURL Apple event handler (r. 2650711).
- Internet Config now includes a system-wide handler for the `'syso' / 'gurl'` Apple event. This allows scripts to access the default systemGURL Apple event handler rather than go through an application's GURL handler (r. 2652051).
- Updated a number of entries in the helper and file types database (rr. 2657695, 2701098, 2720875, 2736928, 2737450, 2751027, 2751714, 2751714).
- Miscellaneous performance and stability improvements have been implemented (rr. 2642736, 2695396, 2734868).

References:

- [Internet Config Documentation](#)

[Back to top](#)

## LDAP Plug-in

The LDAP plug-in provides the interface to allow Directory Services access to multiple LDAP Directory Servers.

- Added default standard type mappings for new record and attribute types to the LDAP plug-in configuration that support `lookupd`. For example, mount record type was added to support NFS mounts (rr. 2738093, 2737923).
- DirectoryService attribute searches using the LDAP plug-in with standard attribute types that mapped to multiple native types now use all of the mapped native types for the search instead of only the first native map (r. 2735224).
- The newly added password plus attribute is now returned with call with `xxxAttributesAll` (r. 2725498).
- Previously the LDAP plug-in was incompatible with the rest of the system with regards to authentication (e.g. `getpwnam`). This was corrected by making the LDAP plug-in return the crypt password when available (r. 2695918).
- Any passwords in the LDAP plug-in configuration file are now properly encrypted and the file is only readable by root (rr. 2685396, 2684378).

[Back to top](#)

## PPP

PPP is a network link layer plug-in that provides access to TCP/IP networks via Point-to-Point Protocol. It supports both PPP over serial and PPP over Ethernet (PPPoE).

- In order to be compatible with certain ISPs, PPP now supports the ability to disable modem error correction and data compression (r. 2650756).
- The OT/PPP API implementation now supports the `OPT_ALERTENABLE` option (rr. 2706455, 2708164).

- PPP now supports the Berkeley Packet Filter (BPF) so that you can use packet analysis tools (for example, `tcpdump`) with PPP connections (r. 2711360). Refer to the `bpf(4)` manual page for more information about BPF.
- PPP can now present dialogs in various circumstances (for example, to ask the user for a password) even when the Internet Connect application is not running (rr. 2563925, 2618282, 2642901, 2654889, 2699228, 2730051).
- The CCL command `CHRDELAY` now works properly (r. 2633622).
- The `pppd` command now loads the PPP kernel extension if it isn't already loaded (such as when you run `pppd` directly from the commandline) (r. 2509083).
- Added some new modem CCLs (rr. 2652691, 2696830, 2697851).
- Improved compatibility with some third party PPPoE implementations (rr. 2643004, 2697540, 2711551).
- Fixed a number of user interface problems with the PPP miniterminal (r. 2691178, 2682180).
- To avoid potential conflicts with other system components, PPP now uses the `PF_PPP` protocol family to communicate between its kernel space and user space systems (r. 2692958).
- Changes to better support PPP over alternative transport mechanisms (for example, `ssh`) have been implemented (r. 2720068).
- Miscellaneous performance and stability improvements have been implemented (rr. 2649995, 2662388, 2658031, 2672347, 2686636, 2692876, 2720279).

[Back to top](#)

## URL Access

The URL Access Manager provides application support for downloading data from or uploading data to a Universal Resource Locator (URL), with support for automatic decompression of compressed files.

- Fixed a bug where calling `URLDisposeReference` would cause a crash on MacOS 10.0.4 (r. 2691455).
- Fixed a bug where URL Access wasn't working with `file:///` URLs (rr. 2664474, 2568588).
- Fixed a problem where URL Access API calls could hang on Dual Processor machines (r. 2649553).
- Fixed a bug where `URLGetURLAccessVersion` was returning the wrong version number (r. 2641384).
- Fixed a problem where trying to download data from certain secure web sites was returning error -6996 when it should not have (r. 2640163).
- Fixed a bug where repeated FTP uploads could hang the machine (r. 2623244).
- Fixed the problem where file downloads weren't given the proper file type and creator codes as specified by Internet Config (r. 2591369).

[Back to top](#)

---

## Security

Authentication, authorization, and cryptographic services on Mac OS X.

### Authorization

The Authorization API lets developers write code to perform system-level tasks, such as installation of files, without requiring root access or special knowledge of the access privilege model in use.

- `AuthorizationCopyInfo` with `NULL` tag no longer crashes (r. 2665498).
- Authorization API is now configurable via the property list file `/etc/authorization` (r. 2611107).
- `Authorization.framework` implementation has moved to `Security.framework` (r. 2608545).
- Canceling the Authorization panel could return the wrong error code `errAuthorizationInternal` (-60008). It now returns `errAuthorizationCanceled` as expected (r. 2665791).

[Back to top](#)

### CSP

Cryptographic Service Providers (CSP) modules are part of the Common Data Security Architecture (CDSA). A CSP provides encryption capabilities for the system. CSP's are either hardware-based (PC add-in card, smart card, etc.), software-based, or a combination of both. A CSP typically provides data encryption and decryption, digital signatures, cryptographic hashing, key generation, random number generation, and secure persistent storage of private keys.

- Fixed an output length error when specifying an encrypt mode with no padding (r. 2668887).

[Back to top](#)

## Keychain

The Keychain provides a secure repository for passwords. Developers can use the Keychain to store or retrieve sensitive data (such as passwords and PINs) when authenticating to services on the user's behalf.

- Fixed a number of problems when displaying or editing keychain items with non-Roman names (r. 2754310, 2745966, 2636246, 2743257, 2721818, 2647584, 2641203, 2632327).
- Keychain Access did not always update correctly when items were added to or removed from a keychain. Now it does (r. 2742442, 2745264).
- `KCMakeKCHandleFromFSSpec` now returns the correct error (`errKCNosuchKeychain`) if a nonexisting file in an existing directory is specified (r. 2610473).
- `KCGetData` now returns `paramErr` if a required parameter is `NULL` (r. 2397162).
- The error `-25300` was being returned from high level Keychain calls when no keychains existed. Now, a user interface is presented to create a keychain instead (r. 2701995).
- Fixed a problem where user interface could be presented even after calling `KCSetInteractionAllowed(false)` (r. 2664401).
- `KCDeleteItem` now works if the item to be deleted is in a keychain other than the default keychain (r. 2626780).

[Back to top](#)

## Security Framework

The framework including CDSA and Authorization APIs.

- Added calls to allow SSL to connect to servers with unknown root certificates (r. 2682603).
- `SecurityServer` now seeds `entropy` to `/dev/random` (r. 2705720).
- Fixed a number of memory leaks in `Security.framework` (r. 2618895).

[Back to top](#)

---

## Tools

This section provides an overview of the changes and improvements provided in Apple's development tools for Mac OS X 10.1.

- A new command line tool, `uuidgen`, that generates UUIDs has been added. This tool is provided as a convenience tool for developers creating plugins (r. 2661967, 2660758).

### `gdb`

`gdb` is the low level debugger for Mac OS X 10.1.

- `gdb`'s `attach` command now allows you to specify a numeric IP address: before the `attach` command would only accept dns host names (r. 2499453).
- The high four bits of addresses were being stripped from addresses displayed in `gdb` when memory translation was turned off (`kdp_trans_off = 1`). This has been corrected (r. 2655351)

References:

- [GDB Release Notes](#)
- [Technical Q&A QA1019, "Can't attach during two-machine debugging with GDB"](#)

[Back to top](#)

### `gcc`

`gcc` is the C, C++, and Objective C compiler for Mac OS X.

- The compiler now respects the `-Wno-long-double` when compiling C++ code (r. 2612261).
- Problems that could occur when declaring default arguments using `extern const` variables have been corrected (r.

2489427).

- A problem that could occur when running `nmedit -s symbols.exp` on a dylib that could result in a non-working, non-debuggable binaries has been corrected (r. 2676317).
- A problem where C++ inlining could conflict with exception cleanup resulting in the compiler generating incorrect code in some rare circumstances has been corrected (r. 2599278).

References:

- [Compiler Release Notes](#)
- [Compiler Tools Release Notes](#)

[Back to top](#)

## Interface Builder

Interface Builder (included on the Mac OS X Developer Tools CD) is Apple's graphical editor for designing user interface components for both Carbon and Cocoa applications. Interface Builder makes creating an application's user interface easier by allowing developers to use its graphical editing environment to manage virtually every aspect of creating a well designed user interface that adheres to the Aqua user interface guidelines.

- Fixed size drawers can now be created in Interface Builder (r. 2743883).
- `NSTextView`s can now be created without visible scrollbars in Interface Builder (r. 2724292).
- The default menu item can now be selected for controls with menus (r. 2723011).
- The inspector for menu items in Carbon nibs now gives you more Carbon Event choices (r. 2715165).
- You can now make connections from custom views inside of scroll views (r. 2712464).
- You can now set the menu title for a standalone menu in a Carbon nib (r. 2708979).
- You can now set command IDs on menu dividers in Carbon nibs (r. 2708810).
- Interface Builder now allows arrow-only popup buttons (r. 2708010).
- `IBInspector`'s `-isResizable` method is now public (r. 2707985).
- Interface Builder now supports the `kPlainWindowClass` and `kAltPlainWindowClass` Carbon window types (r. 2705627).
- Interface Builder now supports the small control version of `NSScroller` (r. 2689920).
- `IBViewResourceDragging.h` is now public. These APIs provide better support for dragging items into other items when using custom palettes (r. 2688073).
- The small control option is now available for Carbon nibs (r. 2682534).
- Interface Builder now sets a default menu item for menus on Carbon bevel buttons (r. 2669684).

References:

- [Interface Builder Release Notes](#)
- [Interface Builder Web Site](#)

[Back to top](#)

## Objective C Runtime

Objective C programs utilize runtime symbol lookup for method calls. This section discusses changes and improvements to these facilities.

- Performance of runtime symbol lookup has been improved and additional routines have been provided for symbol lookup. The new routines include `NSAddImage`, `NSLookupSymbolInImage`, and `NSIsSymbolNameDefinedInImage`.

References:

- [Objective-C++ Release Notes](#)
- `NSModule(3)` man page

[Back to top](#)

## Performance Tools

This section provides an overview of new features and changes provided in the performance tools provided with Mac OS X.

- `MallocDebug` now permits copying backtraces and other information out of the application -- useful for pasting results into a mail message or document.
- `MallocDebug` raises a diagnostic dialog when an application crashes, providing hints about what's happened and why.

- Thread Viewer tool has been added to Mac OS X 10.1.
- Sampler now works better when profiling CFM applications.
- The accuracy of stack backtraces reported by Sampler, sample, and Crash Reporter has been improved.

References:

- [Performance Tools Release Notes](#)

[Back to top](#)

## Project Builder

Project Builder is Apple's integrated development environment (IDE) for Mac OS X. It is designed to fully support all of the major platform initiatives of Mac OS X, such as the Carbon and Cocoa frameworks, Java, and the new application packaging mechanisms. Project Builder provides project editing, search, and navigation, file editing, project building, and debugging facilities for all types of Mac OS X software projects, including applications, tools, frameworks, libraries, plug-in bundles, and kernel extensions and device drivers. It supports the use of C, C++, Objective-C, and Java.

- Class Browser: The Classes tab is now populated with symbols from the indexing information.
- Integrated Documentation Viewing.
- Extended and Improved Symbol Popup: Now uses a quick scanner instead of indexing information.
- Indexing Improvements: Indexing is now on by default, happens during builds, and includes symbols from Java class files and class archives included in the project.
- Optimization level setting has been simplified, a setting is available to control prebinding, two-level name spaces are now supported by the linker, Objective-C++ can now be compiled, by default targets are linked to have a single module.
- Breakpoints are no longer displaced by file editing, startup performance has been improved, CF type and Objective-C object descriptions can now be printed.

References:

- [Project Builder Build Settings Release Notes](#)
- [Project Builder Overview Release Notes](#)
- [Development Environment Release Notes](#)
- [The Project Builder web site](#)

[Back to top](#)

---

## References

Apple's [Mac OS X Developer Documentation Collection](#) on the web.

[Technical Note TN2025, "Mac OS X: versions 10.0.1 through 10.0.4"](#)

[Back to top](#)

---

## Downloadables



Acrobat version of this Note (120K)

[Download](#)

[Back to top](#)