Technical Note TN2010

Mac OS 9.1

This Technical Note discusses changes and corrections in the next generation of Mac OS: Mac OS 9.1. This system follows Mac OS 9.0.4 and contains several enhancements and revised features. All developers interested in creating products compatible with Mac OS 9.1 will want to review this document.

For the most part, Mac OS 9.1 is a system update that fixes bugs and improves operational performance and stability (r. 2469436).

CONTENTS

[Jan 18 2001]

Hardware Requirements

Mac OS 9.1 can be installed on any Macintosh that originally shipped with a PowerPC processor that has at least 32 megabytes of physical RAM and a minimum of 40 megabytes of logical RAM (virtual memory is set to 64 by default if there is less than 64 megabytes of physical RAM installed).

Mac OS 9 1 has been tested and qualified by Apple to run on Apple PowerPC-based Mac OS computers. (Mac OS 9 is not tested or qualified by Apple for use on third party Mac OS-compatible computers.)

Back to top

Installation

Mac OS 9.1 may be installed as an upgrade to Mac OS 7.5.5 and later versions of the Mac OS. The installer program requires Mac OS 8.5 or later to run, and it is recommended that users boot from the installation CD before performing an installation.

Back to top

Finder 9.1

The Finder provides a graphical interface for the user along with other Apple event-based services for applications utilizing the file system. The following lists new features and changes present in Finder 9.1.

Overall changes and fixes

- Get Info windows for applications now allow more free form editing of the Memory partitions. Specifically it is now possible to make changes to either field first without the Finder warning about invalid memory configurations. The Finder will not attempt to apply the changes until the user closes the window (only at this point is a confirmation window presented if the user has entered out of bounds values).

- A contextual menu command has been added that allows users to view the internal contents of Mac OS 9 package folders. The new Show Package Contents contextual menu command allows users to open a package folder and manipulate its contents as if it were any other folder (r. 2406988).

- A Window menu listing the current collection of open Finder windows has been added to the Finder's main menu bar. Selecting an item from the Window menu brings the associated window to the front (or, if the item refers to a pop-up window, then the window will pop-up in to view). The following meta key combinations can be used when selecting items from the Window menu:

  - ○ **no meta key held down** - select & uncollapse
  - ○ **Command** - close (or close drawer)
  - ○ **Command-Shift** - close & put away pop-up
  - ○ **Command-Option** - close others, select & uncollapse this
  - ○ **Control** - collapse others, select & uncollapse this (Single Window)
  - ○ **Control-Option** - uncollapse all, select

- The Finder now supports redirection (by way of the Folder Manager APIs) of the Desktop Folder to volumes other than the startup volume. The Finder now relies entirely on the Folder Manager to determine the location of the Desktop Folder and makes no assumptions about its actual location. (r. 2406346).
- When copying files, it is no longer possible to overwrite a normal document file with an alias file (r.2433219).
- Finder 9.1's user interface no longer allows users to turn on file sharing for folders and volumes with names longer than 27 characters (r. 2257202).
- Editing a file's name inside of the Get Info... window and then locking the file using the check box in the same window no longer reverts the file's name to its last known value (r. 2286023).
- The Finder's Encrypt menu command is no longer available when an alias file is selected (r. 2397515).
- Finder 9.1 will not allow users to bless the desktop folder by moving the Finder and system file to that location (r. 2404745).
- Finder 9.1 does some checking to recognize Mac OS X packages. Namely, if a folder contains the appropriate Mac OS X package information files, then the Finder will identify the folder as a package.
- A crashing problem that could occur when there was an invisible file in the trash and another file with the same name was dragged to the trash has been corrected (r. 2413764).
- Some errors that could occur in the Finder when dragging FireWire drives to trash have been corrected (r. 2515977).
- Pasting custom icon onto a Mac OS X packages is now supported (r. 2538098).
- The Finder will now properly activate windows that it opened in response to Apple events from other applications while running as a background application when the Finder is switched to the front most application (r. 2417802).
- A problem where the Finder could hang when opening a narrow window containing many files has been corrected. This could only happen in non-list-view windows, where the icons do not already have locations assigned to them.
- Finder 9.1 does not allow users to create an alias inside of a drop box. Previous versions of the finder would allow this operation, but they would always fail and leave a partially created alias inside the drop box (r. 2442066).
- Custom icons now display correctly for Mac OS 9 packages (r. 2468468).
- Finder List View redrawing heuristics have been improved (r. 2308818).
- The new Finder no longer crashes when over 200 nested folders are expanded in a list view (r. 2371844).
- Pop-up window tabs are now resized appropriately after the user changes screen resolutions (r. 2372335).
- A problem where folder contents do not update dynamically if the folder is opened during a copy operation has been corrected (r. 2474094).
- Under certain conditions the Command-up-arrow navigation sequence (used to open/select the current selection's parent window) could cause a crash. This problem has been corrected (r. 2515500).
- Under certain conditions it was possible that using the keyboard sequence command-i (get info) followed by the sequence command-delete (move selection to trash) could corrupt Finder memory and cause a crash. This problem has been corrected (r. 2504598).
- the following Keyboard shortcuts have been added so that Finder 9.1 supports the same keyboard commands for keyboard shortcuts used in the Mac OS X Desktop (r. 2525977):
  - ○ **Command-T** - Add selected item to Favorites.

- ○ **Command-Shift-Delete** - empty the trash.

- In some cases, new settings for the memory partition size for an application would not be saved. This has been corrected. (r. 2454019)

- The Finder now clears the ioFlBkDat and ioDrBkDat fields for new files or folders created as the result of a duplicate command or a copy command.

- Items found in the temporary items folder at system startup are, once again, moved to the rescued items folder in the trash (r. 2486192).

> **Note:**
> There is a Tech Info Library article describing this problem located at the address:
> http://til.info.apple.com/techinfo.nsf/artnum/n25134

- The heuristics used by the Finder when identifying text being dragged to one of the Finder's locations as be appropriate for placement in an Internet location file instead of a clipping file have been updated for better compatibility with international text. These routines now require that text used for creating internet addresses contain an explicit url scheme (such as "mailto:", "http://", etc) before an Internet location file will be created as a result of a drag and drop.

> **Note:**
> This change only affects text that contains non-ASCII characters. If the text is pure ASCII, then the same algorithm that was used in Mac OS 9 will be used (r. 2478012).

> **IMPORTANT:**
> It is no longer possible to launch a control panel in the Finder's partition by holding down the Command and Control keys while double-clicking the control panel's icon (r. 2440800).

AppleScript Changes

- A problem, that has existed since Mac OS 8.5, where the 'titled' property of a Finder window was compiled into scripts as the resizable property has been corrected (r. 2472105). For example, the script

```
tell application "Finder" to get the titled of window 1
```

would be compiled as:

```
tell application "Finder" to get the resizable of window 1
```

- In previous versions of the Finder, asking for a list of creator types of all the active processes would return a list of object references instead of a list of four letter strings. This problem appears in Mac OS 8 through Mac OS 9.0.4 and was not present in Mac OS 7.x (r. 1670530). For example:

```
tell application "Finder"
    get creator type of every process
        --> {«class COBA», «class apae», «class 3615», «class Mrr2»,
        «class FMP3», «class ToyS», «class XPR3»}
    beep
end tell
```

This command has been updated so it will always return a list of four character strings as shown here:

```
tell application "Finder"
    get creator type of every process
        --> {"XPR3","FMP3", etc.}
    beep
end tell
```

- The "put away" now has an optional "without asking" parameter (r. 1673117) that can be used as shown in the following listing:

```
tell application "Finder"
    set theDisk to disk "Scratch"
    put away theDisk without asking
end tell
```

The effect of the optional "without asking" parameter is to inhibit any alerts that may appear during processing of the command such as "This disk will re-appear when you next start the machine."

- Scripts, such as the one shown below, attempting to retrieve the names of all ejectable disks no longer fail (r. 2244621). In the past, such scripts would abort with an error such as "Finder got an error: Can't get name of every disk of desktop whose ejectable = true."

```
tell application "Finder"
   name of every disk of desktop whose ejectable is true
end tell
```

- It is now possible for scripts to determine if they are being called as part of the shutdown sequence or the restart sequence (r. 2275596). To discover the current execution state scripts can use the Finder's "execution state" property as shown below:

```
tell application "Finder"
   if execution state is restarting then
      beep 1 -- the machine is being restarted
   else if execution state is starting up then
      beep 2 -- starting up the machine
   else if execution state is running then
      beep 3 -- functioning perfectly
   else if execution state is rebuilding desktop then
      beep 4 -- currently rebuilding the desktop file
   else if execution state is copying then
      beep 5 -- performing a copy operation
   else if execution state is quitting then
      beep 6 -- powering down the machine
   end if
end tell
```

- Since Mac OS 8.5, scripts using modification and creation date properties associated with files did not work (r. 2279739). This problem appears in Mac OS 8 through Mac OS 9.0.4 and was not present in Mac OS 7.x. This problem has been corrected and it is now possible to use scripts such as the following:

```
tell application "Finder"
    set the backup_list to every item of the entire contents of the startup
        disk whose modification date is greater than x
end tell

tell application "Finder"
    set the backup_list to every item of the startup disk whose -
        creation date is greater than the (current date)
end tell

tell application "Finder"
    set the backup_list to every item of the startup disk whose -
        modification date is greater than the (current date)
end tell
```

Back to top

Apple Data Security 2.1

Apple Data Security provides security features to the Mac OS. Security features include facilities for signing and verification of file signatures, password maintenance, and file cryptography.

- Apple Data Security now supports 128-bit encryption and decryption.
- Keychain's handling of http proxy passwords in conjunction with a number of third party applications, and Sherlock, has been improved (r. 2406669, 2391168).

Back to top

Apple File Security

Apple File Security provides cryptographic facilities. Users can use this application to either encrypt a file with a password or decrypt an encrypted file. The Finder's Encrypt menu item is dependent upon the Apple File Security application residing within the Applications folder. Apple File Security does not have the ability to compress and encrypt a folder. This precludes it from encrypting a package.

- Apple File Security 128-bit encryption.
- Apple File Security has been updated for better compatibility with some third party virus protection utilities (r. 2399313).
- A compatibility problem with Multiple Users has been corrected. Here, Apple File Security could mistakenly attempt to encrypt all of the files in a folder containing a file it was asked to encrypt when it does not happen to have read/write permission for the containing folder's parent folder (r. 2442096)

Back to top

Apple Help

Apple Help provides system-wide instructional help services for virtually all aspects of Mac OS.

New Apple Help APIs

- Several new APIs have been introduced with the version of CarbonLib that ships with Mac OS 9.1. The new APIs are outlined as follows.

AHSearch

```
OSStatus AHSearch(
     CFStringRef bookname,
     CFStringRef query);
```

- Tells the Help Viewer to search for the specified query, in the Help book specified by the bookname parameter. If bookname is NULL, search all books. The function is synchronous; it will, if necessary, launch the Help Viewer application, switch to the Help Viewer context, and perform the search.

AHGotoMainTOC

```
OSStatus AHGotoMainTOC(
     AHTOCType toctype);
```

- Tells the Help Viewer to go to the main toc specified by toctype. kAHTOCTypeUser would take the user to the Help Center, and kAHTOCTypeDeveloper would take the user to the Developer Help Center*. The function is synchronous; it will, if necessary, launch the Help Viewer application, switch to the Help Viewer context, and go to the specified table of contents page.

> **Note:**
> NOTE: The Developer Help Center is only supported on Mac OS X. Should a caller attempt to use the kAHTOCTypeDeveloper constant in this API, the Help Viewer will open the Help Center.

AHGotoPage

```
OSStatus AHGotoPage(
     CFStringRef bookname,
     CFStringRef path,
     CFStringRef anchor);
```

Tells the Help Viewer to go to the html file specified by a combination of bookname and path. If bookname is NULL, path must be a full "file://" URL. If bookname is not NULL, path should relative to the book specified by bookname. If an anchor is supplied, Help Viewer will scroll to that anchor in the newly opened file. The function is synchronous; it will, if necessary, launch the Help Viewer application, switch to the Help Viewer context, and go to the specified page.

AHLookupAnchor

```
OSStatus AHLookupAnchor(
     CFStringRef bookname,
     CFStringRef anchor);
```

- Tells the Help Viewer to open the file that contains the anchor specified, and scroll to that anchor. If bookname is specified, the search for files containing that anchor is limited to that book. If bookname is NULL, all books are searched. Anchors should be unique, but if they're not, Help Viewer will display a list of links to all files that contain the specified anchor. The function is synchronous; it will, if necessary, launch the Help Viewer application, switch to the Help Viewer context, and go to the anchor-containing page, or to the list of results.

AHRegisterHelpBook

```
OSStatus AHRegisterHelpBook(
    const FSRef *appBundleRef);
```

- Called by an application each time it launches, this locates a help book that is stored within an application's package, and places an alias to that Help book in the system help location (if not already present), thus enabling the inclusion of that help book in the Help Center the next time Help Viewer is launched.

Help Viewer

Help Viewer is a lightweight HTML rendering application that allows users to browse online help materials.

- A number of memory management and compatibility problems between help viewer, other system facilities, and some third party printers have been corrected (rr. 2535899, 2386521, 2510449, 2525758). The end result of these changes is improved Help Viewer performance and stability.
- Urls of the form help:///full/path/to/file.htm are now understood by the Help Viewer (r. 2537852).
- Problems related to downloading remote content across the Internet, when an alias to the help book is stored in Help book instead of the help book itself, have been corrected (r. 2517221).
- A problem where aliases stored in the help folder would only work if they had the same name as the help book folder has been corrected (r. 2474792). Aliases are no longer required to have the same name.

Back to top

Carbon Support - CarbonLib 1.1

Carbon is the set of programming interfaces for use in Mac OS applications designed to run in both Mac OS 9.1 and Mac OS X. The following new features are present in the version of CarbonLib that ships with Mac OS 9.1. For more information about CarbonLib, please refer to the Carbon SDK

- New Carbon Event model APIs.
- New Carbon Printing model APIs.
- The DataBrowser APIs.
- The new Help Viewer APIs.
- The HTMLRenderingLib APIs.

> **IMPORTANT:**
> Although Mac OS 9.1 ships with CarbonLib 1.1, it is recommended that Mac OS 9.1 users upgrade to CarbonLib 1.2 or later for better performance. At the time of this writing, CarbonLib 1.2 is available as part of the CarbonLib SDK available at the developer web site (http://developer.apple.com/sdk).

Back to top

Inside the ROM Image File

The ROM image file is loaded into RAM at system startup and then that area of memory is set to read only. Contents of the ROM image file replace the older hardware based ROM chip set that shipped with Apple Macintosh computers in the past. Updates to the ROM image file are provided with system updates and with new hardware. The following corrections are present in the ROM image file installed with Mac OS 9.1:

- It was possible for some display colors to be displayed incorrectly after waking up from sleep (r. 2469657). This problem has been corrected.

- The ID number stored in KMAP resources for Pro keyboards (ANSI, ISO, JIS) was not the correct number. This has been corrected (r. 2504668).

Back to top

USB Support 1.4.4

Mac OS USB is built into the Mac OS ROM file, and enables USB support for those Macintosh systems that have USB built in.

- Connected devices are now shown as 'non-removable' in the Name Registry (r. 2439684).
- Better inter-operation with MacsBug has been incorporated into Mac OS USB (rr. 2540060, 2543100,)

Back to top

Inside the System File

The System file contains code and resources for running the computer. This section describes new features, changes, and bug fixes in the System file. The functionality provided by the System file is always present under Mac OS 9.1, even when the shift key is held down at system startup. The system file contains the executable for the libraries listed in this section and resource data used by the system.

- If more than one application is open while the computer was connected to a server and the user pressed the power button, a dialog notifying the user of the impending server disconnection would appear in the background. This has been corrected (r. 2486759).
- If a driver was installed by another driver while processing an accRun call, SystemTask would wind up using stale values for the low memory variables UTableBase and UnitNtryCnt. SystemTask has been updated such that if the driver unit table is moved or resized during an accRun call, it will restart processing the accRun at the beginning of the table (r. 2542196). If you need to install a driver from within a driver during an accRun call in a version of Mac OS prior to Mac OS 9.1, you can do so from inside of a Notification Manager task initiated by your driver while processing an accRun call.
- It was possible for the Process to Process Communication Toolbox to load the Keychain library into some heap other than the system heap and thereby risk having the library close while it was being used. This has been corrected so the keychain library will always be loaded into the system heap by the PPCToolbox (r. 2516363).
- The InterfaceLib glue for the GetFSInfo, SetFSInfo and InformFSM File System Manager calls was trashing register D3. This has been corrected (r. 2522240).
- The File Manager, the Device Manager, and the Delay trap now take advantage of the Multiprocessing API Library's advanced power management support to reduce power consumption and increase battery life.
- Prior to Mac OS 9.1 the limit of the number of outstanding secondary interrupt requests was 40. In Mac OS 9.1 that limit was raised to 256 (r. 2627130).

Back to top

Alias Manager

The Alias Manager is the part of the operating system that communicates with the File Manager to maintain alias records used to store references to 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.

- The following new APIs that take FSRefs instead of FSSpecs have been added to the Alias Manager. These new routines provide the same functionality as the corresponding older Alias Manager calls - the only difference is the use of the new FSRef data type instead of the older FSSpec data type. The new calls are listed below:

```
OSErr FSNewAlias(
    const FSRef *fromFile, /* can be NULL */
    const FSRef *target,
    AliasHandle *inAlias)

OSErr FSNewAliasMinimal(
    const FSRef *target,
    AliasHandle *inAlias)

OSErr FSIsAliasFile(
    const FSRef *   fileRef,
    Boolean     *   aliasFileFlag,
    Boolean     *   folderFlag)

OSErr FSResolveAliasWithMountFlags(
    const FSRef *   fromFile, /* can be NULL */
    AliasHandle inAlias,
    FSRef       *   target,
    Boolean     *   wasChanged,
    unsigned long   mountFlags )

OSErr FSResolveAlias (
    const FSRef *   fromFile, /* can be NULL */
    AliasHandle    alias,
    FSRef      *   target,
    Boolean    *   wasChanged)

OSErr FSResolveAliasFileWithMountFlags(
    FSRef       *   theRef,
    Boolean         resolveAliasChains,
    Boolean     *   targetIsFolder,
    Boolean     *   wasAliased,
    unsigned long   mountFlags )

 OSErr FSResolveAliasFile(
    FSRef       *   theRef,
    Boolean         resolveAliasChains,
    Boolean     *   targetIsFolder,
    Boolean     *   wasAliased)

 OSErr FSFollowFinderAlias(
    FSRef       *   fromFile, /* can be NULL */
    AliasHandle     alias,
    Boolean         logon,
    FSRef       *   target,
    Boolean     *   wasChanged)

 OSErr FSUpdateAlias(
    const FSRef   *   fromFile, /* can be NULL */
    const FSRef   *   target,
    AliasHandle alias,
    Boolean *wasChanged)
```

Back to top

Appearance Manager

The Appearance Manager provides facilities for providing a consistent appearance for graphical user interface elements.

- `DrawThemeGenericWell` has edge case update bug. If an update is needed on a well, and only the outside pixel or two need updating, an optimization within the Appearance Manager will prevent the outside boundries of the well to get drawn (r. 2435805).

Back to top

ATSUI

Unicode is a character set which encodes the characters of many languages in a flat 16-bit number space. ATSUI extends the QuickDraw API and makes it possible for Mac OS applications to draw Unicode text. ATSUI provides both low-level services for drawing Unicode text as well as much of the high-end typographical control previously provided by QuickDraw GX.

- Some third party Unicode cmaps were not being used in ATSUI because the script code for the font was not being read correctly. ATSUI now reads the script code correct so these fonts can be used (r. 2541880).
- The routines `ATSULeftCursorPosition` and `ATSURightwardCursorPosition` now use Unicode Utils routines for cluster movement. This corrects some problems found with the `ATSULeftCursorPosition` and `ATSURightwardCursorPosition` cluster movement operations for surrogates and for Indic clusters (r. 2503643).
- The routines `ATSUPreviousCursorPosition` and `ATSUNextCursorPosition` now pin returned offset to layout boundaries. Previously, these functions could return offsets beyond the layout boundary. For example, if a layout starts at offset 2 but the text from offset 0 begins with a word that is 5 characters long, then setting the initial offset at 3 and asking `ATSUPreviousCursorPosition` to return the previous word boundary would return 0, but now it returns 2 (r. 2503648).
- The routine `ATSUPositionToOffset` is now able to handle surrogate pairs (r. 2497992).
- Under some circumstances when only `QDStyles` were being used, it was possible for `ATSUCompareStyles` to indicate that two style runs were equal when in fact they were not. This problem has been corrected (r. 2485233).
- A problem where Color text printed with ATSUI would not appear as expected on the printed page has been corrected (r. 2393116).
- The Underline character style is no longer applied to the return character (r. 2520184).

Back to top

Drag Manager

The Drag Manager supports Drag-and-Drop operations between windows and applications.

- Under some circumstances, `DrawThemeFocusRect` could disable 2D graphics acceleration. This problem has been corrected (r. 2445982).

Back to top

Deferred Task Manager

The Deferred Task Manager is the part of the operating system that can be used to execute routines with all other interrupts enabled at the end of a hardware interrupt cycle. Normally code running inside of a hardware interrupt will call the Deferred Task Manager to schedule operations that would disable other interrupts for too long if they were processed right away inside of the hardware interrupt cycle.

- It is now possible for a deferred task to call a secondary interrupt handler synchronously from inside of a deferred task using the routine `CallSecondaryInterruptHandler2` (r. 2529860).
- Now, deferred tasks will not run immediately when they are installed from inside of a secondary interrupt handler. Now, when a deferred task is installed during the processing of a secondary interrupt, it will be deferred until the end of the secondary interrupt cycle (r. 2529860).

Back to top

File Manager

The File Manager provides services for storing and retrieving disk-based information. Changes in the File Manager include:

- The performance of allocating additional space to files has been greatly improved.
- Writes to a file that do not begin on a 512-byte boundary can be up to 3 times faster than in previous versions of the File Manager. Specifically, when a File Manager write request does not start on a 512-byte boundary, and that write request extents beyond the end of the 512-byte disk block it began in, and the remainder of that write request will not be cached (either because the `noCache` bit in `ioPosMode` was set, or because the write request was too large to be cached), the cache buffer containing the first part of the write request is immediately flushed to disk. This helps prevent leaving non-contiguous (on disk) dirty cache buffers in the disk cache which must be flushed one block at a time when the file is flushed or closed.
- It is now possible to set the size of the disk cache to sizes larger than 8 megabytes (see the Memory control panel section for more information).
- The File Manager has been revised so that it will no longer process a read command when storing data in the buffer provided would overwrite any low memory globals (r. 2508431).
- `FSGetCatalogInfoBulk` now works correctly with PC-formatted Zip disks (r. 2485093).
- Personal File Sharing is now able to use volumes larger than 4 GB (r. 2283946).
- A problem that could occur when copying large trees of empty folders from ISO 9660 CDs has been corrected (r. 2372435).
- `FSCreateFileUnicode` now works correctly when File Sharing is turned on. Previously, when File Sharing was turned on, files created by this routine would be created with a blank file type and creator even if these values were specified among the parameters to the routine (r. 2397324).
- A problem where a large number of simultaneous asynchronous read or write calls could begin to return incorrect result codes was discovered. Potential circumstances for this problem only existed in Mac OS 9.0 and Mac OS 9.0.4 (r. 2452853).
- A problem where file names could be scrambled when being duplicated on systems with 2-byte character encoding has been corrected (r. 2475992). Specifically, a `FSSetCatalogInfo` call (or its PB equivalents) that didn't try to set the text encoding hint ended up changing the text encoding hint to the current default text encoding. `FSSetCatalogInfo` leaves the text encoding hint unchanged.
- A forced unmount (HUnmountVol) will now close any iterators or "large" files (i.e. files opened with OpenFork) before proceeding with the unmount operation and disposing of the memory structures allocated for the volume (r. 2526761).
- For improved performance, the default allocation block size for volumes over 220GB in size has been set to 8K (r. 2529957).
- Catalog Node IDs (CNIDs) on Mac OS Extended (HFS Plus) volumes can now be reused. If this has happened, bit 12 (`kHFSCatalogNodeIDsReusedBit`) of the VCB's `vcbAtrb` field and the `HFSPlusVolumeHeader`'s attributes field will be set. A disk repair utility that sees this bit set should not complain if it finds that a CNID larger than `nextCatalogID` already in use.

Back to top

File System Manager

The File System Manager provides a general means by which foreign file systems can be installed, identified, and interfaced to the operating system.

- The `GetFSInfo`, `SetFSInfo` and `InformFSM` InterfaceLib glue code was trashing register `D3` in the 68020 emulator. This problem has been corrected and now this register is preserved (r. 2522240).

Back to top

Folder Manager

The Folder Manager provides facilities for locating "special" folders like the Extensions folder in the System Folder without relying on the names of those folders. This aids application localization.

- The "Chewable Items" folders on disks other than the startup disk are now emptied at system startup. In previous versions of the system software only the "Chewable Items" folder on the startup disk was emptied at system startup time (r. 2468800).

- Errors occuring when attempting to create new folders were not being returned by the `FindFolder` routine. This has been corrected and FindFolder now returns any error codes generated when attempting to create a folder (r. 2489334).
- The Folder Manager now does some additional cleanup of network trash folders at mount volume and unmount volume time (r. 2485627).

Back to top

Font Manager

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

- `FMGetFontFamilyTextEncoding` was only returning the base encoding for Japanese PostScript fonts and it was not returning the font variant. This has been corrected and `FMGetFontFamilyTextEncoding` now returns the font variant as well (r. 2489182).

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).

- The routine `GetIconRefFromFile` now returns the correct icon for Mac OS 9 Finder Packages, even when a custom icon has been applied to the package (r. 2481644).
- A problem where 8-bit icons were being used instead of deep ones when the screen was in 16-bit mode has been corrected. Here, the work around was to switch the screen to 32-bit mode (r. 2395259).
- A redundant search for a file named Desktop on volumes containing a file named DesktopDB was eliminated (r. 2388430).

Back to top

Memory Manager

The Memory Manager is the part of the operating system responsible for managing memory allocation requests.

- The MacsBug debugger symbol displayed as the pc location for the memory manager's code has been changed from `BowelsOfTheMemoryMgr` to the more descriptive name `YourHeapIsProbablyCorrupt` (r. 2529682)

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 menu commands are being selected.

- A problem where `IsMenuItemEnabled` would always return true has been corrected. This routine now returns the actual enabled state of the item (r. 2406841).
- A problem where `PopUpMenuSelect` could return an irrelevant menu item, if a keyboard menu command selection was made while a pop-up menu was being tracked with the mouse, has been corrected (r. 2438609).
- The menu bar clock's battery indicator has been made more responsive to changing battery conditions (r. 2497581).

Back to top

MultiLingual Text Editing

MultiLingual Text Editing (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.

- Stability of `TXNNewObject`'s operation under low memory conditions has been improved (rr. 2536146, 2534825).
- `TXNDoFontMenuSelection` now returns an appropriate error code when it is asked to perform an operation with a font that is not supported (r. 2488121).
- The routine `TXNSetDataFromFile` was ignoring the input offsets when called. It now uses these parameters appropriately (r. 2479847).
- MLTE now encodes the a font's full name into encoded files instead of the font's UniqueName (r. 2500541).
- It was possible that justification information would not be used and text would not appear justified as specified in the source file when the text data was read from a RTF format file. This has been corrected (r. 2507443).
- It is now possible to provide `kTXNUnicodeTextData` as the search data for the `TXNFind` routine (r. 2505642).
- The routine `TXNAttachObjectToWindow` now preserves the current GrafPort setting when it is called (r. 2516497).

Back to top

Multitasking/Multiprocessing API

The Multitasking/Multiprocessing Library allows applications to perform preemptive multiprocessing operations on PowerPC computers equipped with one or more processors.

- The routine `MPRegisterDebugger` was not returning an error when an attempt was made to register a debugger for a particular level when one had already been registered for that level. `MPRegisterDebugger` will now return an error code when an attempt is made to register a debugger for a level that has already been registered. Because multiple debuggers may now be registered, other debuggers can be installed while Macsbug is installed (r. 2490379).
- A problem where the routine `MPThrowException` would return a `kInvalidID` error when called with the constant `kMPTaskStoppedErr` has been corrected (r. 2492526).
- A problem where calls to `MPAllocateAligned` would crash after `MPAllocateAligned` was called with the `kMPAllocateNoGrowthMask` mask in the options parameter has been corrected (r. 2510691).
- Memory pools allocated by `MPAllocateAligned` are now properly released when they are no longer needed.
- `MPRemoteCall` takes a new option, `kMPInterruptRemoteContext`, which causes the remote procedure to occur at level-1 interupt time. When the `kMPInterruptRemoteContext` option is used, the `MPRemoteCall` procedure will be called almost immediately (the latency is 1.5 microseconds). This new option allows applications to call interrupt safe `MPRemoteCall` procedures that will execute right away and do not depend on applications calling `WaitNextEvent`.

> **IMPORTANT:**
> `MPRemoteCall` procedures called with the `kMPInterruptRemoteContext` option must be interrupt safe. Hence, they cannot call any routines that may move or allocate memory. For more information about interrupt safe routines, please see Technical Note TN1104, Interrupt-Safe Routines.

- `MPRemoteCall` now calls `WakeUpProcess` when the `kMPOwningProcessRemoteContext` option is specified. This substantially reduces calling latency in `MPRemoteCall`s (from 16 milliseconds to 100 microseconds).
- MP page faults are now resolved at interrupt time instead of system task time. This improves the page fault latency for MP tasks, and eliminates a number of possible blocking situations that can occur during MP thread execution.
- The routine `MPAllocateAligned` could fail even if there was enough memory when the following conditions were met:

  1. The call is being made from a preemptive task created by application 1,

2. Application 1 has not yet called MaxApplZone to make space for the request,
3. The allocation request size is larger than the current free space in the heap,
4. The allocation would fit if the app heap had been grown to full size,
5. The current application is NOT application 1.

This problem has been corrected and the MPAllocateAligned call will not longer fail under these conditions when there is sufficient memory available to fulfill the allocation request (r. 2512273).

- Several bugs in the Multitasking/Multiprocessing API library have been corrected resulting in improved stability in its operation (rr. 2513476, 2520949, 2517072, 2522261, 2524626).

Back to top

Navigation Services

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

- Under certain conditions with inline input, the new folder name field in the 'create new folder' Navigation Services sub-window would not accept text input. This has been corrected (r. 2404367).
- A new item "Connect to iDisk..." has been added to the "Shortcuts" button (r. 2463489).
- An erroneous call to DisposeHandle during the process of adding a favorite to the favorites list in the open dialog has been removed (r. 2470707).

Back to top

PPC Toolbox

The Program-to-Program Communications Toolbox provides process-to-process communications facilities both for processes running on the same machine, and for processes running on different computers connected by either an AppleTalk network or a TCP/IP network.

- The PPC Toolbox now registers itself with Network Services Location. The name registered with NSL is not the 'raw' domain name or IP address, but rather the machine name assigned in the File Sharing control panel. This means that the same machine name will now be seen when browsing by way of both AppleTalk and TCP/IP (rr. 2382164, 2450626).
- It is now possible to change the default port number used by the PPCToolbox for PPC over TCP/IP. The port number used by the PPCToolbox is contained in the 'ppc2' resource in the system file. If this resource is present when the PPCToolbox is being initialized, then the ID number it contains will be used as the default listening port for all incoming PPC over TCP/IP connections (r. 2450615).

  Here is the Rez definition for the default 'ppc2' resource in the system file:

```
type 'ppc2' {
    unsigned integer; // data structure version (1)
    unsigned integer; // default PPCtoolbox TCP port
};


resource 'ppc2' (-16409) {
       1,    // version of resource
       3031  // default PPCtoolbox TCP port
};
```

- It was possible for the location name record describing the remote computer to be set up incorrectly on the listening side of a PPC connection. This has been corrected (r. 2426253).

- Program linking was being turned off after waking up from sleep. Now, if program linking was turned on before the machine went to sleep, it will be turned on again when the machine wakes up (r. 2435676).
- Improvements have been made allowing for more stable operation of the PPCToolbox when used in conjunction with the KeyChain (r. 2516363).

Back to top

Process Manager

The Process Manager shares the processor among multiple applications. Mac OS 9.1 includes the following changes in the Process Manager:

- The Process Manager is now entirely written in native PowerPC code, thus reducing task swapping overhead and improving overall system performance. In particular, performance of calls to `WaitNextEvent`, switching between applications, and Apple events has been improved. Also, there should be a noticeable performance improvement when running on battery power.
- `WakeUpProcess` now properly schedules processes prioritizing them according to their sleep time and their queue of outstanding events.
- In Mac OS 9, it was possible for the system to crash after an application displaying floating windows unexpectedly quit. This problem has been corrected (r. 2540229).
- The Process Manager now makes more efficient use of memory and it coordinates its efforts with the Multitasking/Multiprocessing library (r. 2005897).

Back to top

QuickDraw

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

- `CalcCMask` is now capable of working with PixMaps with long `rowBytes` values (r. 2396796).
- QuickDraw will now correctly draw pictures containing PixMaps with long `rowBytes` values (r. 2358107).
- CopyBits is now capable of scaling down PixMaps with long `rowBytes` values (r. 2407463).

Note:
The Color Manager has been changed so that requests for white and black on 8 bit devices now use the documented requirement that white is the first entry in the pallette and black the last entry. This will cause problems if applications have custom pallettes which do not have white and black in these required positions.

Back to top

QuickDraw Text

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

- If only one QDText accelerator was registered, QuickDraw Text would not notice its presence and text acceleration would not take place (r. 2540704). Hardware acceleration of text rendering is now possible, and when hardware acceleration is not available Velocity Engine based acceleration is used (r. 2447291).

Back to top

Resource Manager

The Resource Manager provides a data file format for storing and retrieving information used during application and operating system run time.

- Some performance enhancements have been introduced that should affect both the performance of the Resource Manager in Mac OS 9.1 and the performance of applications using resource files saved by the Mac OS 9.1 Resource Manager.

  Specifically, the Resource Manager now sorts resource types and ids for files opened read/write, and that the order of resources returned by GetIndResource will likely be different than in previous versions of the Resource Manager. Enumerating across a resource type with GetIndResource while calling AddResource or RemoveResource to add or remove additional resources of that type will likely inconsistent results when compared to the same code using older versions of the Resource Manager.

  And, the number of File Manager Read requests needed to get small resources was reduced from 2 requests to 1 request whenever possible.

- A new Gestalt bit, gestaltRMSupportsFSCalls, has been added to the value returned by the gestaltResourceMgrBugFixesAttrs selector that indicates the presence of the new FSRef calls described in this section.

- New APIs that take FSRefs instead of FSSpecs. These new APIs are described in the following:

OpenResFileFSRef

```
short OpenResFileFSRef(
        const FSRef *ref,
        SignedByte permission);
```

OpenResFileFSRef performs the same operation as the FSpOpenResFile except it allows the caller to refer to the file using a FSRef instead of FSSpec.

CreateResFileFSRef

```
void CreateResFileFSRef(
        const FSRef *parentRef,
        UniCharCount nameLength,
        const UniChar *name,
        FSCatalogInfoBitmap whichInfo,
        const FSCatalogInfo *catalogInfo, /* can be NULL */
        FSRef *newRef, /* can be NULL */
        FSSpec *newSpec) /* can be NULL */

may be NULL
```

CreateResFileFSRef creates a new resource file. It is analogous to the corresponding routine, FSpCreateResFile, but this new routine provides a number of additional parameters for better integration with the new HFS+ APIs.

- **parentRef** - The directory where the file is to be created
- **nameLength** - Number of Unicode characters in the file's name
- **name** - A pointer to the Unicode name
- **whichInfo** - Which catalog info fields to set
- **catInfo** - The values for catalog info fields to set; may be NULL
- **newRef** - A pointer to the FSRef for the new resource file; may be NULL
- **newSpec** - A pointer to the FSSpec for the new resource file; may be NULL

ResourceFileAlreadyOpenFSRef

```
Boolean ResourceFileAlreadyOpenFSRef(
        const FSRef *resourceFileRef,
        Boolean *inChain,
        SInt16 *refNum);
```

`ResourceFileAlreadyOpenFSRef` returns true if the resource file referenced by `resourceFileRef` is already opened. If the file is opened, the `inChain` and `refNum` parameters are set.

- **resourceFileRef** - a reference to the file on disk.
- **inChain** - set to true if the file is open and its map is in the current resource searching chain.
- **refNum** - set to the resource file's reference number. This reference number can be used with other resource manager routines to access the file.

Back to top

Script Manager

The Script Manager provides facilities for display and retrieval of number, date, time, and other information in a language-independent way.

- Some cases were found where `StringToExtended` and `ExtendedToString` were not completely symmetrical. These cases have been fixed (r. 2454794).
- Calls to InitDateCache could cause a crash when the current script's morningStr and eveningStr are empty. This has been corrected (r. 2439135).
- The region name "Flemish" has been changed to "Belgian" (r. 2519932).
- A problem preventing `TransliterateText` from translating Simplified Chinese double byte numbers to ascii numbers has been corrected (r. 2476554).
- The routine `IntlTokenize` was not recognizing the token "<>" as `tokenLessGreat`. Instead it was parsing it as `tokenLess` followed by `tokenGreat`. This has been corrected and now "<>" is recognized as `tokenLessGreat` (r. 2396920).

Back to top

StdCLib 3.7

StdCLib is a shared library providing functionality for programs compiled using the C language. Mac OS 9.1 includes a new version of the StdCLib in the System file's data fork containing the following corrections:

- The major change to StdCLib for Mac OS 9.1 is that all StdCLib functions can now be called from preemptive tasks (r. 2464965). This includes I/O operations in <stdio.h> and memory management in <stdlib.h> (for example, `malloc`, `calloc`, `realloc`, and `free`).

  There are no special initializations which need to be done in order to use this functionality. However, in order to set/get `errno` in <errno.h> from an MP task, the source code must be built with "-d _POSIX_THREAD_SAFE_FUNCTIONS" and the version 3.7 <errno.h> (r. 2464965). This is because `errno` needs to be redefined as a macro which obtains the address of the `errno` value for the task which requests it. That is, `errno` is per-task data--each MP task has its own `errno` value.

  The following global state is managed on a per-task basis as well: the seed for the random number generator `rand` in <stdlib.h>, the state of a scan for `strtok` in <string.h>, and the static data for `ecvt` and `fcvt` in <math.h>, `tmpnam` in <stdio.h>, and `asctime` and `localtime` in <time.h>.

- The following functions have been added to <stdio.h> (r. 2464965):

```
    int getchar_unlocked(void);
    int getc_unlocked(FILE* stream);
    int putchar_unlocked(int c);
    int putc_unlocked(int c, FILE* stream);
    void flockfile(FILE* file);
    int ftrylockfile(FILE* file);
    void funlockfile(FILE* file);

so now one can write...

    flockfile(stdout);
        for (i = 0 ; i < N ; i++)
            putc_unlocked(buffer[i], stdout);
    funlockfile(stdout);

...without incurring overhead on each call
to putc in the loop, and to serialize the output.
```

- To facilitate MP-safeness, the following macros have been removed from the version 3.7 <stdio.h>: `getc`, `putc`, `clearerr`, `feof`, `ferror`, and `fileno`.

Back to top

Text Services Manager (TSM)

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

- Support Carbon event model has been built in to the TSM (r. 2484817). There are no new APIs for this functionality, it is handled entirely by the TSM. Clients of the TSM APIs need only call the existing routines as outlined in Inside Macintosh.

- The `SetTSMCursor` routine is not available in CarbonLib 1.1. Instead, carbon applications use the new `TSMSetInlineInputRegion` and TSM handles setting the cursor for you by way of Carbon events (r. 2483971).

Back to top

Thread Manager

The Thread Manager is part of the operating system providing multi-programming services for use within application programs.

- In Mac OS 9, applications calling DisposeThread on a thread whose stack was allocated in a pre-allocated thread pool, would leak the stacks allocated memory from the pre-allocated thread pool. This was a problem related to the Thread Manager's internal tracking of stack blocks and it has been corrected in Mac OS 9.1 (r. 2410436).

Back to top

Time Manager

The Time Manager provides millisecond and microsecond timing facilities for Mac OS based computers.

- It was discovered that the Extended Time Manager was not drift free when used to scheduling regular interrupts for periods beyond 24 days. This condition has been corrected, and the Extended Time Manager routines are now drift free when used in this way (r. 2471974).

Back to top

Unicode Text Utilities

The Unicode Text Utilities provide a set of APIs for working with Unicode text. These APIs provide services for sorting and collating Unicode text.

- Numbers are now sorted before letters as described in the Unicode technical report (r. 2503166, 2503167).

Back to top

vDSP 1.0

vDSP is a collection of vector and scalar digital signal processing functions that take full advantage of the Velocity Engine used in sound, MP3, speech, audio, video and radar.

- See the forthcoming vDSP SDK for more information. Also, interested developers will want to consult the Velocity Engine web pages at the address http://developer.apple.com/hardware/ve/index.html.
-

Back to top

Virtual Memory Manager

The Virtual Memory Manager provides virtual memory services for the Mac OS.

- Virtual file mapping is now supported. See upcoming technical note for details. Some documentation regarding these routines is currently available in the FileMapping.h header file.
- Page faults caused by MP tasks are now handled with a MUCH smaller latency. With Mac OS 9.1, page faults caused by MP tasks are handled with the latency of an interrupt. Before Mac OS 9.1, page faults caused by MP tasks were handled when SystemTask was called.
- Race conditions between MP tasks running on multiple-processor systems and virtual memory paging operations have been removed. Specifically, modified memory pages being paged out are detached from logical memory before being written to disk, and memory pages being paged in now contain valid data before they are mapped to logical memory.

Back to top

Window Manager

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

- It was possible for `GetWindowRegion` to return an invalid region when the `kWindowDragRgn` selector was provided. This has been corrected (r. 2314117).
- A memory leak that could occur in DisposeWindow for windows created with all of their regions behind the menu bar has been corrected (r. 2369718).
- In the PowerPC version of the Window Manager, `CalcVis` was not setting the `visRgn` of an invisible window to the empty region as was done in the classic implementation of the Window Manager. This has been corrected, and the PowerPC version of the Window Manager now sets the `visRgn` to an empty region for invisible windows (r. 2414578).
- If an application called `HideWindow` on a floating window after the application had been swapped and the floating window had been hidden by the Window Manager, the Window Manager would show the floating window the next time the application was swapped to the foreground. This has been corrected such that now the floating window remains hidden when the application is swapped back into the foreground (r. 2414728).
- The CarbonLib routine `MoveStructureWindow` was not working correctly for collapsed windows. This has been corrected (r. 2423024).
- The font used in Utility window titles now uses the routine `GetApplicationScript` to determine which script to use instead of the `smSystemScript` constant (r. 2424745).
- `IsWindowInStandardState` was not checking the size of the window during calculations. As a result, the `ZoomWindowIdeal` routine did not work correctly with windows positioned partially off screen. This has been

corrected (r. 2478760).

- `StandardAlert` was using the `smSystemScript` constant to select the script for used to set the font in its window. This has been changed such that `StandardAlert` now uses `smCurrentScript` to select the script used to find the font used to display text in its window. With this change, foreign language text will be displayed correctly for localized applications in the `StandardAlert` window when running on a US system with the correct language kit installed (r. 2497250).

Back to top

Control Panels

Control Panels provide user interface facilities for custom configuration of the system.

The Process Manager will choose to launch a control panel file as an application if it contains either a `'CODE'` ID = 0 resource (68K app), or it has a `'cfrg'` resource with a fragment descriptor marked as an "application"type. Control Panels that meet these criteria will not have their `'cdev'` resources called by the system.

> **IMPORTANT:**
> It is no longer possible to launch a control panel in the Finder's partition by holding down the Command and Control keys while double-clicking the control panel's icon (r. 2440800).

Apple Menu Options

The Apple Menu Options control panel provides user interface configuration facilities for the appearance and behavior of the Apple Menu.

- A problem where Apple Menu Options Preferences could be corrupted has been corrected (r. 2528783).
- Apple Menu Options now tracks documents when they are opened using the HFSPlus APIs, and changes to the Apple Menu's hierarchy are now tracked when the HFSPlus APIs are used.

Back to top

Date and Time

The Date and Time control panel provides a user interface for setting the system clock and configuring the display format for both dates and times.

- A problem where after waking from sleep of a duration longer than half an hour, Time Synchronizer would hang when trying to synchronize the clock over an Airport connection. This problem has been corrected (r. 2497293).

Back to top

Extensions Manager

The Extensions Manager control panel provides an enhanced user interface with facilities for developers to provide descriptive information for extensions and to group related extensions together in packages.

- Shareway IP is no longer listed as part of the Mac OS All extension set (r. 2509513).

Back to top

File Exchange 3.0.4

File Exchange provides automatic file name extension to Mac OS file type mapping (for example, mapping the file name extension ".TXT" to the Mac OS file type 'TEXT') along with other services associated with using volumes created for other file systems with Mac OS. The File Exchange control panel allows users to modify settings used by this facility, add new file mappings, mount DOS SCSI disks and drive containers, and set File Translation preferences. New features for this version of File Exchange include:

- The "PC SCSI Disks: Mount at startup" check box is now unchecked by default. This trims 1-1/2 to 2 seconds off the boot time.
- The 'please insert the disk' message was being displayed too often when working with multiple ISO 9660 disks. The reasons for this message appearing have been examined and changed so the number of times it will appear has been minimized (r. 2431666).
- PC DOS formatted volumes containing a folder with the same name as the disk would not copy correctly. Now they do (r. 2475510).
- A problem where the system would hang when mounting some PC formatted floppy disks with USB drives has been corrected (r. 2418015).
- A problem where a copy file operation could stall at the end or become very slow when using a DOS formatted USB hard drive has been corrected (rr. 2333267, 2348961).
- File Exchange did not have a 'sysz' resource for allocating space at system startup, now it does (r. 2373137).
- File Exchange was not managing multiple USB MS-DOS formatted drives correctly. Now it does (r. 2388092).
- A problem where File Exchange could hang Mac OS 9 at system startup time if a SCSI bus was installed that supports 256 target devices has been corrected (r. 2418923).

Back to top

File Sharing

The File Sharing control panel allows for end user configuration of file sharing services on the computer.

- The performance of `PBGetCatInfo` has been improved when file sharing is turned on (r. 2532597).
- It was possible for File Sharing's patches on the file manager to report disk full errors (`dskFulErr`) for file operations on a local volume when the amount of free space on the volume was just over a multiple of 4GB. This problem has been corrected (r. 2536169).
- The new HFS+ routine `FSCreateFileUnicode` was not setting the `kFSCatInfoFinderInfo` when File Sharing was turned on. Now it does (r. 2464870).
- Some potential stability problems related to File Sharing over IP have been corrected (rr. 2488036, 2507440).

Back to top

File Synchronization 1.2

File Synchronization is a utility control panel that allows users to synchronize collections of files and folders.

- Problems resulting in out of memory warnings later leading to crashes have been corrected (r. 2245430).
- A problem where synchronization of folders with custom icons could cause an out of memory error and a crash has been corrected (r. 2245430).

Back to top

General Controls 8.0

The General control panel contains a number of assorted system configuration commands such as the cursor blinking rate, the menu flash count, et al.

- This version of the General control panel is a PowerPC APPC rewritten for better integration with Mac OS and Multiple Users.
- The menu flash and insertion point blinking rate were not always being set appropriately when a user logs in. Now they are (r. 2462891).

Back to top

Internet

The Internet control panel provides a graphical user interface for creating and editing Internet Config preferences, including support for switchable preference sets for handling multiple locations or users.

- Switching the default Web Browser using the pop-up menu now changes both the https and the http helpers. Previously, only the http helper would be changed. Also, switching the default news reader now changes both the nntp helper along with the news helper (r. 2248373).
- It is now possible to choose helper applications that are delivered as packages (r. 2496506).

Back to top

Memory

The Memory control panel provides a user interface for the configuration of virtual memory, RAM disk storage, and the size of the disk cache.

- The maximum disk cache size was increased to 32,736K (32M - 32K); the previous limit was 8160K (8M - 32K). This was done to help performance when allocating large files on large HFS Plus volumes (220GB and larger) initialized with the prior default allocation block size of 4KB.

  For large volumes, the size of the volume's allocation bitmap can end up being bigger than the disk cache, causing the File Manager to do a huge amount of I/O when allocating space for a file. Setting the size of the disk cache to a value larger than the size of the bitmap allows the bitmap to remain in the cache, dramatically speeding up allocation requests.

  When a volume uses 4KB allocation blocks (which was the default for large volumes initialized with Mac OS 9.0.4 or earlier), the allocation bitmap will be 32KB per GB of volume size. Volumes over 220GB initialized with Mac OS 9.1 default to an 8KB allocation block size, or 16KB of bitmap per GB of volume size. For example, when using a 200GB volume, the disk cache should be GREATER than 6400K; the next larger size is 6432K. When using a 400GB volume initialized with Mac OS 9.0, you should set the disk cache size to 12832K or larger for best performance.

  Note that the default disk cache size is always less than 8MB (8192K), for maximum compatibility.
- Added a new `Gestalt` selector, `gestaltDiskCacheSize` (`'dcsz'`), that returns the size of the disk cache size in bytes.
- Increased the speed that the disk cache size changes when the mouse button is held down for a long time on the up or down arrows.
- The computations for maximum disk cache size were changed to better take into account physical RAM needed by ROM-in-RAM, built-in video, RAM disk, and the OS. This makes it much harder to set the disk cache and RAM disk so large that you don't have enough physical RAM free to boot. The maximum size is larger when VM will be on than when VM will be off (since the OS uses less physical RAM when VM is on). The logical (VM) RAM size also affects the maximum disk cache size; increasing VM can increase the maximum disk cache size in some configurations.
- Added a `SysError` early in boot if the RAM disk setting is too large for the system to boot. This can happen if VM was supposed to be on but was turned off (for example, by booting with the shift key down), after removing RAM, or if a driver uses a large amount of physical RAM. The RAM disk will be turned off and you will be forced to restart.
- The correct free disk space is now displayed in the Virtual Memory panel. It was previously pinned to the maximum VM size.

Back to top

Remote Access

The Remote Access control panel provides user interface facilities for configuring PPP and ARA connections.

- Remote Access now supports Challenge-Handshake Authentication Protocol allowing remote connection services to verify a PPP connection periodically both during the connection establishment and after the connection has been established (r. 2297793).
- PPP now reports its correct MTU size to upper protocol layers for better support of PPPoE type connections (r. 2483604).
- A problem with the remote access dialog enter key handling code that could result in a crash if the key was typed in rapid succession has been corrected (r. 2536881).
- A problem where the modem would not dial if the option to connect automatically when starting TCP/IP applications was on and DialAssist was on has been corrected (r. 2541002).
- A problem where the Modem and Remote Access control panels could wind up with generic icons after an upgrade install has been corrected (r. 2532014).
- A problem where a powerbook PB101 would crash if the lid was closed while ARA was disconnecting has been corrected (r. 2385200).
- A problem where in areas with ten-digit dialing, DialAssist would dial the long distance access number before the prefix has been corrected (r. 2419739).
- A problem preventing long CCL scripts from functioning correctly has been corrected (r. 2458391).

> **IMPORTANT:**
> AppleScript's Remote Access Commands are not installed with Mac OS 9.1 as they are not compatible with Remote Access 4.0. As a result, Remote Access Commands will not be available to script writers in Mac OS 9.1.
>
> It is recommended that script writers using any of the commands provided by Remote Access Commands re-write their scripts to use Network Setup Scripting instead. Examples can be found in the Open Transport (Networking) AppleScript Guidebook module, which can be downloaded from http://www.apple.com/applescript on the Web (rr. 2572457, 2530257).

Back to top

Software Update

Software Update provides a way to find and install the latest system software for your computer by way of the Internet.

- Update now works better when Multiple Users is turned on (r. 2388398).
- Many of the error messages displayed by Software Update have been revised so they are more descriptive and helpful (r. 2452634).
- Software Update will now correctly stop the USB Software Locator when the user cancels (r. 2465100).
- Software Update now works correctly across firewalls (r. 2476559).
- A call to `InitWindows` was removed from the Software Update Engine background only application (r. 2513784).

Back to top

Startup Disk

The Startup Disk control panel provides user interface facilities for choosing the disk that will be used to run the computer the next time it starts up.

- Mac OS 9 comes with an entirely new Startup Disk control panel written entirely in PowerPC code and delivered as an APPC style application. The new control panel features better integration with the rest of the system and improved user interface facilities.
- With Multiple Users turned on, it is no longer possible for users other than the owner to have access to the Startup Disk control panel (r. 2496511).

Back to top

Web Sharing

Web Sharing provides facilities for users to configure and set up the "Web Sharing" folder on their computer to act as a World Wide Web site. Web Sharing allows users to publish documents on an intranet or the Internet. Documents placed in the Web Sharing folder will be available to other users by using means of any web browsers available on any platform.

- A problem where web sharing would not run if the TCP/IP control panel was not present (but TCP/IP was enabled and loaded) has been corrected (r. 2501286).

Back to top

System Extensions

System extensions are located in the Extensions folder inside the System Folder. For a system extension to be active, it must be located in the Extensions folder when the system starts up. Extensions are not loaded if the Shift key is held down while the system is starting up.

AppleScript 1.5.5

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

Mount Volume uses Keychain

- As of AppleScript 1.5 and later, the 'mount volume' scripting addition will look in the Keychain to see if there is a password for the requested server. Note that keychain passwords are stored according to the connection protocol: a key for a server accessed via AppleTalk is different than a key for accessing the same server through TCP/IP. So if you've created a key for that server connected via TCP/IP, you must use the TCP/IP form of 'mount volume' (using a URL of the form afp://host.domain.com/volume) in order to have the keychain key work correctly. Similarly, if you created the key while connected to the server via AppleTalk, you must use the AppleTalk form of 'mount volume' to use that key.

Rounding 'as taught in school'

- In AppleScript 1.5 and later, the 'round' scripting addition now can take an additional enumerated parameter: 'as taught in school'. With this parameter, numbers whose fractional parts are less than .5 are rounded towards zero, and fractional parts of .5 and above are rounded away from zero.

Unicode Support

- In AppleScript 1.3, initial support was added for Unicode data. AppleScript could send, receive, load, and store Unicode strings, and convert between text strings and Unicode strings. It could not display them or perform any operations on them. In AppleScript 1.5 and later, Unicode strings can be displayed in script editors (if the appropriate fonts are installed in the system). CAUTION: In AppleScript 1.5.5, certain operations on Unicode text can cause AppleScript to crash or hang. Apple advises that if your script is handling Unicode text, you should convert the Unicode text to a string before attempting to manipulate the text in any way.

- Unicode support in AppleScript 1.5 and later has been improved to correctly convert Unicode strings of mixed languages to the appropriate language-specific characters if the necessary Language Kits are installed.

Mac OS X Support

- In AppleScript 1.4.3, an option was introduced to save scripts as "Mac OS X applets". Due to changes in Mac OS X, these do not actually work on that system. As of AppleScript 1.5 and later, applets saved as "Mac OS X applets" require Mac OS X to run.

- AppleScript version 1.5 and later can now handle script files in which the script data is stored in the data fork, rather than the resource fork. All scripts that ship with Mac OS X will be stored in the data fork, but the Script Editor by default will continue to save script data in the resource fork for compatibility with previous versions of AppleScript.

- As of AppleScript 1.5, AppleScript and the Script Editor recognize files that end in '.scpt' as script files, if they are missing other type and creation data. This should make it easier to transfer script files using media or communications methods that do not retain Mac type and creator data.

- AppleScript 1.5 and later now accept all styles of line endings for the purposes of finding paragraph breaks in strings. The expression 'every paragraph of s' will break paragraphs correctly whether the string is Mac-style (paragraphs end with 0x0D), UNIX style (lines end with 0x0A), or DOS style (lines end with 0x0D0A).

Other Changes

- The Script Editor in AppleScript 1.4.3 introduced a "proxy icon" in the title bar of script windows: an icon that represents the saved script file on disk. Unfortunately, dragging that icon to a disk folder did not move the script file to that folder; in AppleScript 1.5 and later, it does.

- In Script Editor 1.4.3 and earlier, command-clicking on the title of a script window that is not front most would bring up a pop-up menu for that script's path. Users expect command-dragging the titles of background windows to drag those windows without bringing them to the front; in Script Editor 1.5 and later, that is indeed what happens.

- In AppleScript 1.5 and later, the "Where is application <name>?" dialog box now uses Navigation Services instead of Standard File.

- In AppleScript 1.5 and later, the "info for" scripting addition now returns the size of the file or folder as a real number, not an integer. This accommodates files or folders that are greater than 2 Gigabytes in size.

- In AppleScript 1.4.3 and earlier, certain arithmetic operations on odd integers were handled differently from the same operations on even integers, making those operations approximately half as fast. Odd integers are now treated the same as even integers, so certain repetitive arithmetic operations on odd integers will be faster in AppleScript 1.5 and later.

- AppleScript documentation erroneously reports the effective range of AppleScript integers: the true range is 2^29-1 to -2^29 (i.e. 536870911 to -536870912). In AppleScript 1.3 through 1.4.3, integer addition and subtraction handled arithmetic operations at the extremes of this range incorrectly; for example, 536870911 + 1 would return -536870912. In AppleScript 1.5 and later, operations on large-magnitude integers will be promoted to reals, so for example 536870911 + 1 returns 5.36870912E+8.

- In AppleScript versions 1.4.3 and earlier, asking for the middle item of an empty list (e.g. 'middle item of {}') would cause AppleScript to crash. This has been fixed in 1.5 and later.

- In AppleScript 1.4.3, running multiple applets simultaneously could occasionally cause AppleScript to crash. This has been fixed in AppleScript 1.5 and later.

- In AppleScript 1.4.3 and earlier, providing large numbers as the 'seed' parameter to the 'random number' scripting addition would have undesired effects: similar large numbers would consistently produce the same initial random value (though subsequent values would be more randomly distributed). This is especially troublesome because AppleScript itself uses the system clock as a seed value if none is provided, and most system clock values tend to be similar large numbers. In AppleScript 1.5 and later, large seed values produce more evenly distributed initial values.

- In AppleScript 1.3 through 1.4.3, on rare occasions the 'random number' scripting addition, when given real numbers for its range, would generate a number fractionally larger than the upper bound of the range. In AppleScript 1.5 and later, all random numbers are assured to be no greater than the upper bound.

- In AppleScript 1.4.3 and earlier, attempting to access item 0 of a list (e.g. 'item 0 of {2, 3, 4}') would produce no error and no result. In AppleScript 1.5 and later, it produces an 'index out of range' error.

- In AppleScript 1.4.3 and earlier, creating a script with no 'run' handler (explicit or implicit), but having a 'parent' property of a script with an implicit run handler, would work in the Script Editor but crash if the script were saved as an applet. For example:

```
script crasher
	property parent : current application
	display dialog "This hasn't crashed"
end script
property parent : crasher
```

Note the script has no statements other than the script and property definition, so there is no 'run' handler; when

AppleScript tries to run it, it looks in its 'parent' script ('crasher') and tries to run its run handler. But because there is no explicit run handler in the 'crasher' script, it tries to get the run handler from its parent, and loops infinitely. This has been fixed in AppleScript 1.5 and later.

- In Folder Actions 1.4.3 and earlier, Folder Actions would create an invisible "Icon\n" file in any folder you control-clicked, whether you attached a Folder Action to it or not. As of Folder Actions 1.5.5, the file is created only if you actually attach a folder action.
- In Folder Actions 1.5.5 the 'action name' and 'action number' parameters of the 'remove action from' and 'edit action of' events have been changed to 'using action name' and 'using action number' for readability. Older scripts should de-compile to the new form automatically.

Back to top

AppleShare Client 3.8.8

AppleShare Workstation Client provides file sharing services for computers networked with file servers.

- The AppleShare Workstation Client no longer uses Apple Shared Library Manager (ASLM).
- A crashing bug that could occur with a custom User Authentication Module that uses volume passwords has been corrected (r. 2546186).

Back to top

ATAPI CD/DVD Driver

The ATAPI CD/DVD Driver provides support for connected ATA CD and DVD devices.

- Problems where directories on DVD-RAM disks could get corrupted when switching disks have been corrected (r. 2545573).

Back to top

ColorSync 3.0.3

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

- Changes have been made in the user interface to ensure the user can only pick gamma values that are within the range that can actually be used by a display. Specifically, the gamma calibrator will no longer allow selection of negative gamma calibration values (r. 2496933).
- The ColorSync Profiles installed with Mac OS 9.1 have been updated to the current ICC standard (r. 2505195).
- Alpha channels are now preserved in calls to CWMatchBitmap where a progress proc has been specified (r. 2318181).
- CMLaunchControlPanel now works correctly (r. 2384459).
- ColorSync now requires a valid version field in profile headers as specified in the ICC Profile Format Spec (r. 2404863).
- Requests to copy path based profiles with CMCopyProfile would always fail with an error -43. This problem has been corrected (r. 2411802).
- It is now possible to set the model number property of a profile by way of AppleScript commands. These commands existed before, however they did not work as expected (r. 2417435).

Back to top

FireWire Support 2.7

FireWire is Apple's Mac OS support for the IEEE 1394 High Performance Serial Bus standard. This extension adds services to the Mac OS that support the use of third-party FireWire hardware and software.

- A crashing problem that could occur when Unplugging/Replugging devices during QuickTime movie Playback has been corrected (r. 2520111).
- A problem where a third party application could hang on wake from sleep when a number of FireWire devices were connected has been corrected (r. 2452899).
- Some older DV camcorders do not send video correctly when connected to a FireWire hard disk drive. Customers using Final Cut Pro or iMovie to record video onto a FireWire hard disk drive may have experienced lost video frames when using such cameras. FireWire 2.7 will make older cameras send video correctly when a FireWire hard drive is present. (rr. 2408186, 2404727, 2398097).
- FireWire now supports multiple matching data in SBP-2 drivers. If the symbol TheDFMDescriptorSize is present, multiple matching data will be evaluated. This allows a driver to precisely match several different devices, without using range matches that could pick up unwanted additional devices. If used, TheDFMDescriptorSize should have a value equal to the number of matching data sets present - such as 2. For an example. see the file FWSBP2DiskDriver.c in the FireWire SDK.

> **Note:**
> If a driver with multiple matching data is loaded by FireWire 2.6 or an earlier version, only the first matching data will be used.

- Some third party FireWire drivers re-use their command object when calling `FWWaitForDeviceReplug`. This is illegal and causes a hang. As a convenience to our customers, in this specific case only, we now check for improper command object re-use and tolerate it.
- A problem where a hang would occur as a result of a retry counter not being decremented has been corrected (r. 2323090).
- Changed `LynxFWIMGetCycleTime` to not try to read PCI registers when the PCI bus might be powered off or PCI Memory Space might not be enabled (r. 2452899).
- Drivers can now pin their DFM scores to a maximum value by exporting a table of maximum scores called `MaxDFMScores`. There is one unsigned 32 bit entry for each entry in `TheDFMDescriptor` and its value is the maximum score desired for that entry. This allows drivers to exactly match on some values but still generate a low score and thus allow other parties to override it (r. 2526813).
- FireWire will no longer send write responses to broadcast writes on OHCI systems (as specified in 1394).
- FireWire now implements the `BUS_TIME` register correctly, as required by 1394.
- A problem where bug where transmit isoch timestamps could be off by 4 seconds on Uni-N systems has been corrected.
- All OHCI timestamps now have the full 7 bits of seconds (Previously only 3 bits were provided).
- A problem where FireWire could hang on wake from sleep when both a firewire hard drive and a firewire camera were connected has been corrected (r. 2452899).

Back to top

HTMLRenderingLib

HTMLRenderingLib is Apple's lightweight HTML rendering engine capable of displaying HTML 3.2 files.

- Applications are now passed all url types in their optional URL redirection routine installed by calling HRRegisterNewURLUPP. Previously, http type URLs were handled by the HTMLRenderingLib and were never seen by the URL redirection routine. Now, http type URLs will only be handled by the HTMLRenderingLib when either no URL redirection routine has been installed or when the installed URL redirection routine returns hrURLNotHandeldErr (r. 2418857).
- Calls to the HTMLRenderingLib would sometimes leave the GrafPort's origin or clip region in an unknown state. This no longer occurs (r. 2406856).
- It is now possible to call the HTMLRenderingLib from stand-alone code resources (r. 2459981).
- The routine `HRUtilGetURLFromFSSpec` was not adding a trailing '/' character when constructing URLs referring to

directories. Now it does (r. 2407002).

- The Cursor now changes to a hand when it is positioned over links (r. 2502044).
- Visited links were not being re-colored appropriately under some circumstances. This has been corrected (rr. 2309910, 2484091).
- A problem where bullets would overlap in ordered lists when the HTML source contained unrecognized tags has been corrected (r. 2362018).
- A problem that could occur when attempting to display a frames document when all of the frames have the same name has been corrected (r. 2430596)
- `HRGetRenderedImageSize` was returning the vertical size of an image, minus two pixels. now, it returns the correct size (r. 2468994).
- Rendering a large table with missing closing TD tags could cause a crash. This has been corrected (r. 2514177).

Back to top

IrDA

The IrDA extension provides an infrared communication link between print drivers and IrDA-capable printers.

- A problem where a crash could occur when performing several consecutive print jobs has been corrected (r. 2512225).

Back to top

Mac OS Runtime for Java

Mac OS Runtime for Java (MRJ) is Apple's implementation of the Java runtime environment. MRJ supports running Java applications, and enables Mac OS applications that use MRJ's JManager API to embed Java applets and other Java content.

- JAR files are now read in a more efficient and memory conscious way (r. 2001604).
- Applet Runner now uses Navigation Services to select files when running on systems where Navigation Services is available (r. 2287732).
- WHWindow.cp's destructor is now virtual (r. 2421498).
- A problem where Java would crash when a force quit has been corrected (r. 2472481).
- A null pointer exception that could occur when closing a window has been corrected (r. 2485969).
- To assist in localization the following awt.properties have been added (r. 2487549):

```
#The name of the Java Console window
AWT.console.name=Java Console

#Java Console Edit Menu item names
#These names should be localised.
AWT.editmenu.title=Edit
AWT.editmenu.undo.name=Undo
AWT.editmenu.cut.name=Cut
AWT.editmenu.copy.name=Copy
AWT.editmenu.paste.name=Paste
AWT.editmenu.clear.name=Clear
AWT.editmenu.selectall.name=Select All

#This is the font and size used inside the console window.
#This should be left as is.
AWT.console.fontname=Monaco
AWT.console.fontsize=9
```

- Problems redrawing large images have been corrected (r. 2490722).
- `FileInputStream` will now set it's member variable `fd` to `NULL` when it can't open a file (r. 2508818).
- MRJ now allocates a larger memory pool the first time `malloc` is called (r. 2510365).
- A problem where Applets with startup errors could crash Apple Applet Runner has been fixed (r. 2511642).

Back to top

Multiple Users 1.3

Multiple Users is a multi-user software extension that provides the capability for a number of users to access a single computer. Each user logs in to her own environment that provides her own preferences and some file security.

- Better support for networking environments (Macintosh Manager) has been added.
- New support for KidSafe is now included.

Back to top

Network Services Location (NSL)

Network Services Location (NSL) allows you to register and search for services on a network in a protocol-independent fashion. New features for NSL include:

- Network browser was inappropriately displaying the message "name server not responding" in some circumstances. This problem has been corrected (r. 2406166).
- The routine `NSLStandardGetURL` was not appropriately disposing of all of the thread manager tasks it had allocated. Now it does (r. 2424817).
- A problem where NSLStandardRegisterURL could crash under low memory conditions has been corrected (r. 2429189).

Back to top

Open Transport 2.7.4

Open Transport is the networking technology for Mac OS 9.1.

- Open Transport no longer uses ASLM.
- A problem where a data block could be reused even though it had a positive reference count has been corrected. This problem was most apparent when using Ethernet in promiscuous mode (r. 2459754).
- It was possible for a crash to occur when switching the TCP configuration from BootP to DHCP. This has been corrected (r. 2471989).

Back to top

OpenGL 1.2

OpenGL for Macintosh is Apple's implementation of Silicon Graphics' industry-standard 3D graphics API.

- Additional Velocity Engine optimizations have been added including Velocity Engine support in the compiled vertex array path.
- OpenGL can now draw anti-aliased lines and points.

- Support for TCL drivers has been added.
- The following extensions have been added to OpenGL:
  extensions
  - GL_ARB_texture_env_add, GL_EXT_abgr, GL_EXT_bgra, GL_EXT_texture_env_combine, GL_EXT_clip_volume_hint, GL_EXT_rescale_normal, GL_NV_texgen_reflection, GL_EXT_texture_lod_bias, GL_EXT_SGIS_texture_edge_clamp, GL_EXT_abgr, GL_EXT_bgra, GL_EXT_texture_lod_bias, GL_EXT_SGIS_texture_edge_clamp, GL_ARB_transpose_matrix, GL_ARB_texture_compression, GL_APPLE_packed_pixels, GL_ARB_texture_env_add, GL_EXT_abgr, GL_EXT_bgra, GL_EXT_texture_env_combine, GL_EXT_clip_volume_hint, GL_EXT_rescale_normal, GL_NV_texgen_reflection, GL_EXT_texture_lod_bias, GL_EXT_SGIS_texture_edge_clamp, GL_ARB_transpose_matrix.
  - Now supported in the software renderer:

    GL_EXT_abgr, GL_EXT_bgra, GL_EXT_texture_lod_bias, GL_EXT_SGIS_texture_edge_clamp.
- Support for non-coherent memory in the compiled vertex array path has been added.
- It is now possible to add fog in the compiled vertex array path.
- ARGB32 textures in 16-bit color are now supported.
- Features improved support for OpenGLProfiler library.
- TCL support at the GLD layer is now enabled.
- OpenGL 1.2 standard is now in the interfaces.
- GL_ARB_texture_compression and GL_APPLE_packed_pixels are now included in the interfaces.
- LRU/MRU texture paging is now supported with ATI hardware.
- Several optimizations to texture upload path have been added.

Back to top

PlainTalk

PlainTalk provides both speech recognition services and services for the translation of text into audible speech. The Mac OS 9 installation ships with English Text-to-Speech 2.0.2, Plaintalk Speech Recognition 2.0, and Mexican Spanish Text-to-Speech 2.0.2. English Text-to-Speech is installed as part of the default Mac OS 9 installation. In Mac OS 9.1 English Localized versions, Plaintalk Speech Recognition 2.0 is installed as part of the standard install. For other localized versions of Mac OS 9.1, it can be installed as a custom install option. On some localized OS releases, individual Plaintalk components, may be moved to the CD Extras folder on the System Software CD or removed entirely from the OS.

- A problem where the effects of the "Make this application speakable" Speakable command would not be recognized until speakable items was quit and restarted has been corrected (r. 2496982).

Back to top

QuickTime 4.1.2

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

Mac OS 9.1 includes an improved version of the Digital video (DV) codec. Improvements in this version of the DV codec provide significantly better performance and quality over previous versions. These changes should be of benefit to all QuickTime developers, especially developers using video editing applications.

- High-quality decompression has been optimized for both Power Mac G3 and G4 computers. This includes improved speed for rendering in r408 and v408 formats, improved speed for playback to YUV accelerated windows, and improved speed for high-quality, single field video.
- The quality the rendered images produced by normal (low) quality decompression on Power Mac G4 computers has been

improved.

- The quality of the compression in medium and best-quality cases for Power Mac G3 and Power Mac G4 computers has been improved.
- Multiprocessor support has been enabled for vector decode (i.e.,high quality, low quality,playback and scrub/render). Multiprocessor support has also been enabled for encode in the vector case. For the actual DV operation, the results are nearly two times faster on a two CPU computer than the same code running in non-multiprocessor mode.

Back to top

Time Synchronizer

Time Synchronizer is a faceless background application that synchronizes the computer's clock over the Internet.

- Time Synchronizer could hang if the location stored in PRAM could not be found in the location database (r. 2499437).
- Alerts explaining error conditions detected at startup preventing the Time Synchronizer from running were not being displayed (r. 2500473).

Back to top

URL Access 2.1

URLAccess is a shared library providing high-level Internet APIs. This facility allows clients to download or upload data from or to the Internet. With this API, all applications can become Internet-enabled with just a few additional calls.

- URL Access now supports with SSL Ref v3, 128-bit encryption.

Back to top

USB Printer Sharing 1.0.1

It allows users to share their USB printers over TCP/IP (physical or wireless).

- Mac OS 9.1 ships with USB Printer Sharing 1.0.1.

Back to top

Fonts

Fonts provide information used in drawing characters to the screen and to other graphics devices.

- A LineFeed cmap entry has been added to all fonts installed with Mac OS 9.1 (r. 2487954).
- The sidebearings from the kashida glyph were removed from the Arabic fonts to allow better justification of Arabic text in Textension (r. 2343333).
- The priority of the kashida character was changed in the Arabic fonts so it will be used appropriately in justification (r. 2503404).
- Korean System Fonts now support Unicode (r. 2444584).

Back to top

Apple Menu Items

Items in the Apple Menu Items folder appear in the Apple Menu.

Sherlock 2

Sherlock provides search facilities for Mac OS computers. In addition to the traditional capabilities of the Find application, Sherlock supports searching by content and Internet searches.

- The forward delete key now works as expected in Sherlock (r. 2457688).
- Inline input now works correctly in the Save Custom Settings dialog (r. 2498288).
- Sherlock now uses user@host rather than an email address for a password when establishing an anonymous FTP connection (r. 2463751).
- The "Open Enclosing Folder" now works correctly (r. 2383272).

Back to top

Apple System Profiler

The Apple System Profiler (ASP) provides a simple way to obtain information regarding the current system configuration. Information provided by the Profiler is useful for customer support representatives in assisting users.

- ASP now displays the CarbonLib version.
- ASP is now capable of reading reports generated by the ASP found in Mac OS X.
- Now correctly displays a PowerBook G3's modem as 33.6 and the protocol as "V.34".
- ASP no longer displays "VRAM" size on machines that do not have built-in VRAM/video cards
- ASP features improved facilities for identifying memory configurations on iBooks, and for identifying PC66 memory.
- ASP now matches FireWire drives to the FireWire nodes.
- ASP features improved facilities for displaying USB related information including:
    - Some issues related to finding USB buses have been corrected.
    - USB Printer Sharing support
    - ASP was describing USB "serial" printers as type "serial". ASP now describes USB serial printers as type "USB".
- ASP's AppleScript support has been improved in the following ways:
    - ASP would crash when processing requests asking specifically for System Folder information and nothing else.
    - ASP would fail when trying processing requests asking specifically for devices and volumes information and nothing else.
    - ASP wouldn't return Power Mac G4 (AGP Graphics) serial numbers. Now it does.
    - ASP didn't return mulitple user's name/environment or video note. Now it does.
    - Scripting terminology has been changed as follows:

```
"AtEase" is now "At Ease"
"MultipleUses..." is now "Multiple Users..."
"StartupDisk" is now "Startup Disk..."
"...logicboard..." is now "...logic board..."
```

- ○ Fixed TCP/IP comments that referenced AppleTalk.
- ○ The Unique logic board ID returned by ASP now returns a different number that is more unique than the previous set of ID values returned.

- ASP no longer marks slower RAM, as many users were interpreting this as a "bad" thing, while all it really means is that the computer runs at the speed of the slowest RAM (a minor speed difference). ASP continues to show the speed of the RAM, to allow users to buy matching RAM speeds, if they choose.

Back to top

Utilities

Utilities include items useful for maintaining your computer. Normally, these items are placed in the Utilities folder on your startup volume when you install your system software, but some of them can be found in the "Utilities" folder on the Install CD.

Disk First Aid

Disk First Aid is an application program that provides disk diagnostic and repair services. It is normally located in the Utilities directory in the startup disk's root directory.

- DFA detects and replaces code in an HFS Plus "wrapper" for volumes that were initialized with Mac OS 9.0.x. This prevents a crash or hang early in boot when the boot volume has lots of files or directories (typically 100,000 or more).
- Now DFA refuses to repair when logged in as a Panel user and the volume has open files.
- DFA will not complain if a Mac OS Extended (HFS Plus) volume indicates certain text encodings are in use, even when they aren't actually in use. These will be updated if any other repairs are performed. This is most likely to occur when language kits have been used (r. 2406117).
- Disk First Aid might mistakenly identify a Mac OS Extended (HFS Plus) volume that was re-initialized as a Mac OS Standard (HFS) volume as a Mac OS Extended volume and report that is unable to repair the volume. This problem has been corrected (r. 2509928).
- When writing a volume header or alternate volume header, the signature and version fields are always set to their correct values.
- DFA will not generate an error if the next CNID is on volumes containing no files or folders.
- DFA will now properly check the space used by a startup file. This fixes an incorrect error message about some Mac OS X startup volumes (r. 2455546).
- A possible off-by-one error in checking the next CNID has been corrected.
- When repairing an orphan directory thread record, DFS will now start searching at the first leaf node. Previously, some nodes could have been skipped, causing the repair to fail (r. 2496023).
- When repairing a volume with open files, DFA now copies the parallel Extents data to the parallel Extents FCB, and not to the parallel Catalog FCB. Also, DFA copies HFS Plus extent information to the ForkControlBlock if available, or to parallel FCB (r. 2412846).
- DFA no longer generates an error regarding the next CNID if an HFS Plus volume has had its CNIDs reused.
- When trying to figure out the volume type, DFA now uses the alternate volume header before the main volume header.
- DFA can now accommodate larger B-trees (r. 2405114).
- DFA no longer generates bad error message on volumes containing large numbers of long filenames (r. 2404773).

Back to top

Drive Setup 2.0.3

Drive Setup provides facilities for updating hard disk drivers and for formatting hard drives.

- Problems with the visual appearance of the partition selection box with some drives have been corrected (r. 2392640).
- Drive Setup will not allow the drive to spin down if Test disk is running on DVD-RAM media (r. 2451144).

Back to top

Apple Extras

Apple Extras include a number of items placed in on your hard drive during system installs. These items may include utilities, tools, or programs meant for specific purposes.

CD Audio Player

The CD Audio player is a utility that allows you to play audio CDs.

- A problem where the CD Audio Player could trash some LowMem Globals has been corrected (r. 2502234).
- The CD Audio player will no longer show the data session as an extra track at the end of track list when a Multisession enhanced audio CD is inserted (r. 2509403).
- Problems playing and displaying track times for Multisession enhanced audio CDs have been corrected (rr. 2510627, 2510686).

Back to top

World Text

An MLTE based text editor that allows Unicode text editing.

- New MLTE based text editor found in the cd extras folder on the cd.

Back to top

MacsBug 6.6.3

MacsBug is Apple's low-level debugger, providing a user interface for the disassembly and perusal of machine-level instructions.

- MacsBug 6.6.3 should be used with Mac OS 9.1 (r. 2540060).
- Even better bus error's magic address now points to a no-access page in memory. Any attempts to either read or write to this address will enter MacsBug with and EBBE error.

Back to top

Downloadables

Back to top

Technical Notes by API | Date | Number | Technology | Title
Developer Documentation | Technical Q&As | Development Kits | Sample Code