

Technote 1142

Mac OS 8.5

Being a compendium of revised and new features of interest to software developers in release 8.5 of the Mac OS.

This Technote discusses changes and corrections in the next generation of Mac OS: Mac OS 8.5. This system follows [Mac OS 8.1](#) and contains several new and revised features including:

- more PowerPC code
- consolidated control panels
- faster system startup time
- improved AppleScript performance
- new Unicode text drawing facilities
- new HTML-based help facilities

Contents

- [Hardware Requirements](#)
- [Installation](#)
 - [Install Mac OS](#)
- [Finder](#)
 - [Autorouting and Extended Routing](#)
 - [New Finder Display Facilities](#)
 - [Internet Location Files and Clipping Files](#)
 - [AppleScript Support](#)
- [Apple Help](#)
 - [Apple Guide](#)
 - [Help Viewer](#)
- [Inside the System File](#)
 - [General](#)
 - [Alias Manager](#)
 - [Appearance Manager](#)
 - [ATSUI](#)
 - [Code Fragment Manager](#)
 - [Control Manager](#)
 - [Device Manager](#)
 - [Dialog Manager](#)
 - [Display Manager](#)
 - [Disk Initialization Package](#)
 - [Drag Manager](#)
 - [Driver Services Library](#)
 - [Event Manager](#)
 - [File Manager](#)
 - [File System Manager](#)
 - [Folder Manager](#)
- [Control Panels](#) (Continued)
 - [ColorSync](#)
 - [Control Strip](#)
 - [Date & Time](#)
 - [File Exchange](#)
 - [File Sharing](#)
 - [Internet](#)
 - [Keyboard](#)
 - [Location Manager](#)
 - [Memory](#)
 - [Monitors & Sound](#)
 - [Remote Access](#)
 - [TCP/IP](#)
 - [Users and Groups](#)
 - [Web Sharing](#)
- [System Extensions](#)
 - [Apple CD/DVD Driver](#)
 - [Apple Enet](#)
 - [AppleScript](#)
 - [AppleShare Workstation Client](#)
 - [Application Switcher](#)
 - [Desktop Printing](#)
 - [Find By Content](#)
 - [Internet Config](#)
 - [LaserWriter 8](#)
 - [Mac OS Runtime for Java](#)
 - [Network Setup Extension](#)
 - [OpenTransport](#)
 - [PlainTalk](#)
 - [QuickTime](#)

- [Font Manager](#)
 - [Help Manager](#)
 - [Icon Services](#)
 - [InterfaceLib](#)
 - [List Manager](#)
 - [MathLib](#)
 - [Memory Manager](#)
 - [Menu Manager](#)
 - [Navigation Services](#)
 - [Notification Manager](#)
 - [PC Card Manager](#)
 - [PPC Toolbox](#)
 - [Process Manager](#)
 - [QuickDraw](#)
 - [QuickDraw Text](#)
 - [Script Manager](#)
 - [Serial Driver](#)
 - [Sound Manager](#)
 - [Standard File](#)
 - [Text Services Manager](#)
 - [Time Manager](#)
 - [Translation Manager](#)
 - [Virtual Memory Manager](#)
 - [Window Manager](#)
 - [Control Panels](#)
 - [Appearance](#)
 - [Apple Menu Options](#)
 - [QuickDraw 3D](#)
 - [Apple Menu Items](#)
 - [Apple System Profiler](#)
 - [Key Caps](#)
 - [Network Browser](#)
 - [Sherlock](#)
 - [Apple Extras](#)
 - [Apple Video Player](#)
 - [SNMP Administrator](#)
 - [Utilities](#)
 - [Disk Copy](#)
 - [Disk First Aid](#)
 - [Drive Setup](#)
 - [Apple Developer Utilities](#)
 - [MacsBug](#)
 - [Macintosh Programmer's Workshop \(MPW\)](#)
 - [Items No Longer Supported](#)
 - Appearance Extension
 - Apple Remote Access Client
 - Cyberdog
 - Desktop Pictures control panel
 - Mac OS Easy Open
 - Map control panel
 - OpenDoc
 - Open Transport/PPP
 - WorldScript Power Adapter
-

Hardware Requirements

Mac OS 8.5 can be installed on any Macintosh that originally shipped with a PowerPC processor that has at least 16 megabytes of physical RAM and a minimum of 24 megabytes of logical RAM. Also, Mac OS 8.5 can be installed in the Mac OS X Server DR2 Blue Box by way of standard install.

Mac OS 8.5 is tested and qualified by Apple to run on Apple PowerPC-based Mac OS computers. (Mac OS 8.5 is not tested or qualified by Apple for use on clone Mac OS-based systems.)

Installation



Install Mac OS 8.5

This is an improved version of the "Install Mac OS" application provided with Mac OS 8.0 and 8.1. This new "Install Mac OS" application offers an improved user experience by allowing the user to customize components before any installation begins and not requiring further user intervention until the installation is complete.

The "Install Mac OS" application program requires System 7.0 or later to run. Typically, users will boot from the installation CD.

The "Install Mac OS" application will update a System Folder whose version is 7.1 or later. The installer will perform a "clean" installation onto volumes that contain earlier versions of the system software (or no system software at all).

New and improved features include:

- File & Target Lists -- a more compact way of storing file copy commands to get around Resource Manager limitations.
- Alias Atom -- an atom for creating aliases during installations.
- Enhancements to the Installer Engine AE suite.

Related Materials:

- The [Installer SDK 1.2](#)
-



Finder 8.5

Finder 8.5 offers improved window redraw performance, faster file copies, and several other new, improved, or extended user interface elements. Specific additions include:

- A new dialog allows users to repair broken aliases.
- Using contextual menus, an alias of any Finder object can be added to the "Favorites" folder.
- Using contextual menus, AppleScript scripts triggered by various Finder actions can be attached to any folder.
- Sounds now play in response to many user actions.
- Finder's resource fork is now read-only. In many cases, this will prevent accidental corruption of Finder.
- The system stability warning dialog box is now displayed when Finder is re-launched after a system error.
- It is now once again possible for an application to specify 'alis' as a file type in its bundle resources to indicate that unresolved alias files should be sent to its "open documents" Apple event handler.
- Developers can designate certain files as "busy" files. Busy files are files that are in the process of being created or copied, regardless of whether File Manager has them open. The Finder will treat any file whose type is set to zero or to a type within the range 'bzy ', 'bzy!', ..., 'bzy?' as a busy file and will not attempt to change its type or creator. The range 'bzy ', 'bzy!', ..., 'bzy?' is provided for developers wishing to animate a file's icon during its creation.

COMPATIBILITY NOTE

Archive developers should store the entire contents of both the `FInfo` and `FXInfo` records and the `DInfo` and `DXInfo` records when archiving files and directories. All fields in these records should be preserved, including flags and fields marked as reserved (with the exception of the "inited" bit for files containing desktop database information).

COMPATIBILITY NOTE

Although `frFlags` and `fdFlags` are located at the same offset in catalog records, some of the flags now have different meanings when the catalog record refers to a file versus when the catalog record refers to a directory. Developers should not assume flags that are only defined for files are unused when the catalog record refers to a directory. See the flag descriptions in `<Finder.h>` for more information about individual flags.

Autorouting and Extended Routing

- Finder now exclusively uses Folder Manager for auto-routing items dropped into the System Folder (see the [Finder Interface Reference](#) chapter of *Inside Macintosh: Mac OS 8 Toolbox Reference* for more information).
- Finder attempts to utilize extended routing information stored in the resource fork of a file if the file's extended routing info bit is set in its Finder flags. When such files are dropped into the System Folder, Finder will attempt to read a 'rout' (ID=0) resource, which should contain a list of routing information entries. Typically, there will be only one entry in the list. When the list contains more than one entry, Finder walks through each entry and uses the last one that matches.

The 'rout' resource entries are defined as follows:

```
typedef struct
{
    OSType creator;           // set to zero
    OSType fileType;         // set to zero
    OSType targetFolder;      // set to 'macs'
    OSType destinationFolder; // the folder to route into
    OSType reserved;         // set to zero
}
RoutingResourceEntry;
```

The creator, fileType, and reserved fields should be set to zero.

When the file is dropped into a folder of the type specified in targetFolder, it will be routed to the folder specified in destinationFolder. Currently, only the System Folder is supported for the targetFolder field, so this field should be set to the System Folder's FindFolder ID ('macs'). Example: if the file should be routed to the "Preferences" folder, the destinationFolder field should contain 'pref'.

Before the Finder will look for the 'rout' resource in a file, the routing information flag in the ioFlXFndrInfo.fdxFlags must be turned on. Developers can access this flag using the kExtendedFlagHasRoutingInfo mask defined in <Finder.h>.

COMPATIBILITY NOTE

Some of the fields documented in FInfo, DInfo, FXInfo, and DXInfo no longer accurately reflect how these fields are actually used by the system. Now, the accurate definition is provided by the FileInfo, FolderInfo, ExtendedFileInfo, and ExtendedFolderInfo data structures, respectively. In particular, they contain the definition of where to find the routing bit.

New Finder display facilities

- The Get Info window can now display multiple panels of information.
- Finder preferences are now consolidated into a single multiple panel window.
- A new control has been added to the View Options window, allowing users to revert the current view to a standard view.
- List view columns are now resizable.
- The new Finder can display badged icons.
- Support for proportional scroll bars has been added.
- Proxy icons are displayed in window title bars.
- View font and size is now set in the Appearance control panel.
- To override the default message displayed by the Finder when a file cannot be opened, developers may include a 'STR' ID = -16397 resource in any file's resource fork. This message text will be displayed when the user attempts to open a file that cannot be opened, overriding the Finder's default message.
- In previous versions of the Finder, files with their custom icon bit set that did not contain a suite of custom icons were not handled correctly. The Finder now resets the custom icon bit and uses the default icon for these files.

COMPATIBILITY NOTE

Starting with Finder 8.5, the invisible "Icon<cr>" file placed in directories may contain resource-based information used by the Finder other than an icon suite. It should only be deleted if it contains no resources at all.

- The Finder now uses the new `gestaltVMInfoType` Gestalt selector to determine how information about virtual memory will be displayed in the "About This Computer" window. Developers wishing to control how the Finder displays virtual memory information may redefine this selector to return any of the values shown in Table 1. Table 1 describes the various virtual memory information display modes available in this version of the Finder.

Table 1. Finder virtual memory display modes determined by the value of the `gestaltVMInfoType ('vmin')` selector.

Selector Name	Value	VM Information Displayed
<code>gestaltVMInfoSizeStorageType</code>	0	Display either the string "virtual memory on" or the string "virtual memory off", the size of the backing store, and the name of the volume where the store is located (the default).
<code>gestaltVMInfoSizeType</code>	1	Display either the string "virtual memory on" and the size of the backing store or the string "virtual memory off".
<code>gestaltVMSimpleType</code>	2	Display either the string "virtual memory on" or the string "virtual memory off".
<code>gestaltVMNoneType</code>	3	No virtual memory information will be displayed, even when virtual memory is turned on.

Internet Location Files and Clipping Files

- Finder supports enhanced text clipping naming. Text clipping file names are now generated using the first few characters in the text supplied as the text clipping (truncated using the ellipsis character as appropriate). Previous Finders would name text clippings as 'text clipping', 'text clipping 1', and so on. For example, the new Finder may name a text clipping generated using this sentence as "For example, the...". This new facility supplements the clipping file naming facilities introduced with MacOS 8.0 (see the ["Controlling Clipping File Names"](#) section of Technote [TN1102](#), "Mac OS 8" for more information).
- Both Internet location files and clipping files are created by the Finder when it receives a Drag-and-Drop command. The type of file created depends on the contents of the drag as described below.
- Clipping files created by the Finder will have one of the types listed in Table 2. All clipping files have a creator type of `kClippingCreator ('drag')`.

Table 2. Clipping file types created by Finder.

Constant Name	OSType	Description
<code>kClippingPictureType</code>	<code>'clpp'</code>	a QuickDraw picture (<code>'PICT'</code>)
<code>kClippingTextType</code>	<code>'clpt'</code>	contains ASCII text (<code>'TEXT'</code>)
<code>kClippingSoundType</code>	<code>'clps'</code>	contains a sound resource of type <code>'snd '</code>
<code>kClippingUnknownType</code>	<code>'clpu'</code>	unknown data

Finder will create an Internet location document when it receives a drag containing an Universal Resource Locator (URL). A drag containing an URL will be noticed by the Finder in one of two ways:

- If the drag contains data of flavor type 'url', then the data will be treated as an URL.
- If the drag contains data of flavor type 'TEXT', then Finder will examine the text data attached to the flavor and determine if it is an URL.

When it finds URL information, Finder will create an Internet location file assigning the file type according to the URL type. Table 3 lists the file types the Finder will assign to Internet location files. Internet location files created by the Finder will have a creator type of kInternetLocationCreator ('drag').

Table 3. Internet location file types assigned by the Finder.

Constant Name	OSType	Description
kInternetLocationHTTP	'ilht'	an http address
kInternetLocationFTP	'ilft'	a ftp server address
kInternetLocationFile	'ilfi'	a file on disk
kInternetLocationMail	'ilma'	an email address
kInternetLocationNNTP	'ilnw'	news group or article
kInternetLocationAFP	'ilaf'	Apple Filing Protocol address
kInternetLocationAppleTalk	'ilat'	an AppleTalk address
kInternetLocationGeneric	'ilge'	some other resource

COMPATIBILITY NOTE

Specific file formats and contents of both clipping files and Internet location files are undocumented and subject to change without notice.

AppleScript Support

- Additions to Finder's AppleScript support include:
 - terminology for new features
 - new view options
 - attach and remove actions from folders
 - list actions attached to a folder
 - execute a handler when a folder is opened, closed, added to, has items removed from it, or has its window moved or resized
 - new support for scripting existing features
 - query and change information about any file or folder
 - open, close, and reposition windows and icons
 - query information about running processes
 - query or set sharing privileges
 - move files to the Trash
 - eject disks
 - restart or shut down the machine
 - create folders or alias files

Related Materials:

- The [AppleScript](#) section
 - The [Icon Services](#) section
 - Technote [TB09](#), "Finder Flags"
 - Technote [TB42](#), "Finder Icon Positioning and File Initialization"
 - Technote [TB535](#), "Finder Q&As"
 - Technote [TN1126](#), "Checklist for Building Applications and Extensions"
 - [Inside Macintosh: Icon Services \(preliminary draft\)](#)
 - The [Finder Interface](#) chapter of [Inside Macintosh: Macintosh Toolbox Essentials](#)
 - The [Finder Interface Reference](#) chapter of [Inside Macintosh: Mac OS 8 Toolbox Reference](#)
 - [Inside Macintosh: AppleScript Finder Guide](#)
 - Anderson, G. [Scripting the Finder](#). [Develop](#) 20
 - The [AppleScript SDK](#)
-



Apple Help

Apple Help provides system-wide instructional help services for virtually all aspects of Mac OS. The Apple Help technology has two main components: Help Viewer and Apple Guide.

Apple Guide 2.3

In Mac OS 8.5, Apple Guide is used to coach users through sequences of commands required to perform a task. Apple Guide 2.3 provides the following features:

- supports calling a specific Apple Guide sequence by name from AppleScript
- backwards compatibility with older guide files
- integrated with Help Viewer

Help Viewer 1.0

Help Viewer is a lightweight HTML-rendering application that allows users to conveniently browse online help materials. Help Viewer provides the following facilities:

- Uses the Apple Information Access Toolkit (AIAT) with the installed help files. The results are displayed in a dynamically generated HTML page.
- AppleScript support allows callers to search for any topic or string and display the results.
- URL links can be used to run AppleScript scripts.
- URL links can be used to open Apple Guide sequences.

Apple recognizes many developers are interested in more information, but no further developer-level information is available at this time.

Related Materials:

- The [Help Manager](#) section
 - The [Help Systems](#) Technote Collection
 - The [Help Systems](#) Q&A Collection
 - [Inside Macintosh: Apple Guide Complete](#)
 - AppleGuide web pages at <http://www.apple.com/macos/apple-guide/>
 - The [Apple Guide 2.1.1 SDK](#)
-



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 8.5, even when the shift key is held down at system startup.

General

- In Mac OS 8.5, the emulated 68K processor runs in user mode even when VM is off. All 68K interrupt code runs in supervisor mode. As a result, all 68K interrupt code utilizes the interrupt stack pointer instead of the user stack pointer (the stack used by non-interrupt code). This change reduces the chance of the stack overrunning the application's heap zone during interrupt processing.

WARNING:

Developers should not assume the emulated 68K processor is running in supervisor mode when Virtual Memory is OFF.

Developers who require access to supervisor mode 68K instructions can use the `_EnterSupervisorMode` trap to switch the processor mode. The assembler code shown in Listing 1 illustrates the correct way to call the `_EnterSupervisorMode` trap and enter supervisor mode. Here, the result returned by `DebuggerGetMax` is examined to determine if `_EnterSupervisorMode` (selector 8) is available. If it is available, `_EnterSupervisorMode` is called.

Listing 1. The correct way to ensure the 68K processor is in supervisor mode.

```

; Is _DebugUtil available?
; Code running on a IIci or later can skip this step.
    move.w    #_Unimplemented,d0
    _GetToolTrapAddress
    move.l    a0,-(sp)
    move.l    #$A08D,d0          ; get _DebugUtil
    _GetToolTrapAddress
    cmpa.l    (sp)+,a0          ; was this previously
                                ; unimplemented?
    beq.s     @noEnterSupervisorMode

; Is the _EnterSupervisorMode selector available?
    _DebuggerGetMax              ; check how many _DebugUtil
                                ; selectors are implemented
    cmpi.l    #8,d0             ; is selector 8 available?
    blt.s     @noEnterSupervisorMode ; EnterSupervisorMode isn't
                                ; available

    _EnterSupervisorMode        ; switch to supervisor mode
                                ; (switches stacks)
    move.w    d0,-(sp)          ; and save the previous SR contents
                                ; on the stack
    bra.s     @inSupervisorMode

; If _EnterSupervisorMode wasn't available, then put
; sr on the stack
@noEnterSupervisorMode
    move      sr,-(sp)          ; save the current SR

@inSupervisorMode

; ...
; do stuff in supervisor mode
; remember that the stack has been swapped if we switched
; from user mode to supervisor mode.
; ...

    move      (sp)+,sr          ; restore the SR

```

COMPATIBILITY NOTE

DebuggerGetMax is available on all machines supported by Mac OS 8.5. On other systems, developers must test for the _DebugUtil trap before calling DebuggerGetMax. Testing for VM to determine if the _EnterSupervisorMode trap should be used is considered a programming error: developers should use the above method instead.

fromFile, if not NULL, is used to resolve relative aliases.

alias is the AliasHandle to resolve.

target contains a pointer to an FSSpec record that will refer to the file or directory referred to by the alias when the function completes successfully.

wasChanged refers to a Boolean variable that will be set to true if the alias record has been modified by ResolveAliasWithMountFlags.

mountFlags can be set to kResolveAliasFileNoUI to prevent any user interaction, including disk switch alerts, while the alias is being resolved.

The routine ResolveAliasWithMountFlags is identical to ResolveAlias with the exception that it provides the mountFlags parameter, allowing callers to suppress disk switch alerts.

ResolveAliasFileWithMountFlags

```
OSErr ResolveAliasFileWithMountFlags(FSSpec *theSpec,
                                     Boolean resolveAliasChains,
                                     Boolean *targetIsFolder,
                                     Boolean *wasAliased,
                                     unsigned long mountFlags);
```

theSpec on input refers to a FSSpec record that refers to an alias file. If the function completes successfully, it will refer to the file or the directory that was referred to by the alias file.

resolveAliasChains turns on or off resolution of chains of alias files (for example, an alias file that refers to an alias file that finally refers to the target file).

targetIsFolder points to a Boolean variable that will be set to true if the target referenced by the alias file was found to be a folder.

wasAliased returns true if theSpec referred to an alias file.

mountFlags can be set to kResolveAliasFileNoUI to prevent any user interaction, including disk switch alerts, while the alias is being resolved.

The routine ResolveAliasFileWithMountFlags is identical to ResolveAliasFile with the exception that it provides the mountFlags parameter, allowing callers to suppress disk switch alerts.

COMPATIBILITY NOTE

Before calling IsAliasFile, ResolveAliasWithMountFlags, or ResolveAliasFileWithMountFlags, developers must determine if these routines are available by testing the Alias Manager gestalt value's gestaltAliasMgrResolveAliasFileWithMountOptions bit. Also, see the InterfaceLib section for important information about linking with the new InterfaceLib.

- The routines `NewAliasMinimal` or `NewAlias` could cause heap corruption if an invalid `FSSpec` record with the length of the name field greater than 63 bytes was provided as a parameter. These routines now return a `paramErr` error when they receive an invalid `FSSpec` record.

Related Materials:

- The [Code Fragment Manager](#) section
- Technote [FL30](#), "Resolving Alias Files Quietly"
- Technote [FL505](#), "Alias Manager Q&As"
- The "Updating aliases when File Sharing state changes" Q&A in Technote [NW515](#), "AppleShare Q&As"
- The "QuickTime alias and FSSpec system services under System 6" Q&A in Technote [QT510](#), "Movie Toolbox Q&As"
- The Alias Manager section in Technote [TN1041](#), "Inside Macintosh: Files Errata"
- Technote [TN1083](#), "Weak-Linking to a CFM-based Shared Library"
- The [Alias Manager](#) chapter of [Inside Macintosh: Files](#)
- The [Using Aliases](#) section in the [Finder Interface](#) chapter of [Inside Macintosh: Macintosh Toolbox Essentials](#)
- The [Alias Manager](#) section (page 2-64) in the [Movie Toolbox](#) chapter of [Inside Macintosh: QuickTime](#)

Appearance Manager 1.1

The Appearance Manager provides facilities for providing a consistent appearance for graphical user interface elements. The Appearance Manager is not to be confused with the Appearance extension, which is compatible only with previous versions of Mac OS, and provides updates to Control Manager, Dialog Manager, Menu Manager, and Window Manager, as well as providing the 1.0.X versions of Appearance Manager. Appearance Manager 1.1 is delivered as part of the Mac OS 8.5 System File and includes the following new features:

- Switchable visual appearances.
- Support for data-driven appearances (though no appearance files are included with Mac OS 8.5).
- Many new theme brushes and text colors.
- Drawing primitives for most interface elements.
- Support for saving and restoring the `GrafPort` state regardless of whether the current appearance draws using a color or pattern.
- Several different styles of scrollbars and check boxes.
- Appearance and font change notifications are provided by way of Apple Events. Each event uses `kAppearanceEventClass` as its class and one of the constants listed in Table 4 as its ID. There are no parameters to any of the events, and these events are sent only to those processes that have called `RegisterAppearanceClient` to register as an Appearance Manager client. Table 4 lists the new Apple Events sent to Appearance Manager clients.

Table 4. Appearance and font change Apple Events. These events use the Apple Event class `kAppearanceEventClass ('appr')`.

Constant Name	OSType	Description
<code>kAEAppearanceChanged</code>	<code>'thme'</code>	appearance changed
<code>kAESystemFontChanged</code>	<code>'sysf'</code>	system font changed
<code>kAESmallSystemFontChanged</code>	<code>'ssfn'</code>	small system font changed
<code>kAEViewsFontChanged</code>	<code>'vfnt'</code>	views font changed

- Animated cursor support.
- Support for sound during various user interactions.
- An Appearance folder has been added to the System Folder. The Appearance folder is used as a common location to store Appearance-related data files, and it is also the location of the Theme Files, Sound Sets, and Desktop Pictures folders. Appearance files are autorouted to their appropriate folders when dropped into the System Folder. Table 5 lists the new Appearance folders present in Mac OS 8.5.

Table 5. New folders for Appearance and their `FindFolder` selectors.

Folder Name	OSType	Description
Appearance	<code>'appr'</code>	Appearance-related materials
Theme Files	<code>'thme'</code>	location for theme files
Sound Sets	<code>'snds'</code>	location for appearance-related sound sets
Desktop Pictures	<code>'dtpf'</code>	location for desktop picture files. Files of type <code>'JPEG'</code> are auto-routed into this folder when dropped into the System Folder.

- The file `<Folders.h>` contains the constants listed in Table 5 along with their symbolic names (`kAppearanceFolderType`, `kThemesFolderType`, `kSoundSetsFolderType`, and `kDesktopPicturesFolderType`) for use in calls to `FindFolder`.

Related Materials:

- The [Control Manager](#), [Dialog Manager](#), [Menu Manager](#), and [Window Manager](#) sections
- Q&A [TB39](#), "Appearance Versions"
- The [High Level Toolbox](#) Technote Collection
- Technote [TN1102](#), "Mac OS 8"
- Technote [TN1121](#), "Mac OS 8.1"
- The [Finder Interface Reference](#) chapter of *Inside Macintosh: Mac OS 8 Toolbox Reference*
- The [Appearance Manager Reference](#) chapter of *Inside Macintosh: Mac OS 8 Toolbox Reference*
- The [FindFolder](#) description in the [Finder Interface](#) chapter of *Inside Macintosh: Macintosh Toolbox Essentials*
- The [Responding to Apple Events](#) chapter of *Inside Macintosh: Interapplication Communication*
- *Inside Macintosh: Macintosh OS 8 Human Interface Guidelines*
- The [Appearance 1.0.3 SDK](#)

ATSUI (Apple Type Services for Unicode Imaging)

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.

- Provides text imaging services for 16-bit Unicode (UTF-16).
- ATSUI fully conforms to The Unicode Standard, Version 2.1.
- ATSUI fully supports the Unicode bidirectional algorithm, including the bidirectional ordering codes.
- ATSUI text-drawing can be integrated within a QuickDraw-only application (unlike QuickDraw GX, ATSUI does not require developers to adopt a non-QuickDraw API for all text drawing).
- For fonts which lack a Unicode ' cmap ' table, ATSUI will automatically generate one "on the fly". The font file is not modified, but the conversion makes certain assumptions about character encodings. For best results, it is recommend that Japanese fonts include a Unicode ' cmap '.

Related Materials:

- Technote [TN1102](#), "Mac OS 8"
- Technote [TN1121](#), "Mac OS 8.1"
- *Inside Macintosh: Text Encoding Conversion Manger*
- The [Text Encoding Converter 1.3 SDK](#)
- The [ATSUI SDK](#)

Code Fragment Manager

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

- Code Fragment Manager calls to `ResolveAliasFile` at system startup would attempt to put up the "Please Insert The Disk" alert before any drawing environment was established. CFM now calls `ResolveAliasWithMountFlags` to disable the disk switch alerts.
- The Virtual Memory Manager's file mapping code in Mac OS 8.1 makes large chunks of file-mapped memory resident that are certain or very likely to be used in the near future. In Mac OS 8.5, that functionality has been moved into the Code Fragment Manager. Application launch performance should be approximately the same in most cases. However, files that have multiple code fragments (for example, fat CFM 68K/PowerPC applications) may benefit from this refinement.
- CFM provides an optional facility where an application can specify a special folder to be searched for shared libraries. A field in the 'cfrg' resource provides the resource ID of an 'alis' resource. If this field is non-zero, CFM attempts to read the resource from the application's resource file and resolve it. (The assumption is that this is a relative alias to a subfolder of the application folder -- for example, a repository for plug-ins.) The Code Fragment Manager now calls the new Alias Manager routine `ResolveAliasWithMountFlags` to resolve aliases referenced by the 'cfrg' resource to prevent any user interaction while libraries are being loaded.
- The Application Support folder is now part of the CFM search path.
- CFM defines a new bit in the 'cfrg' resource which forces the fragment's code into the application heap. See the constant `kCFragLibUsageMapPrivatelyMask` in `<CodeFragments.h>` for more info. [Radar 2205428]

Related Materials:

- The [Alias Manager](#) section
- Technote [TN1083](#), "Weak-Linking to a CFM-based Shared Library"
- Technote [TN1127](#), "In Search of Missing Links"
- The Code Fragment Manager section in Technote [IM_ERRATA 05](#), "Inside Macintosh -- PowerPC System Software Errata"
- The [Code Fragment Manager](#) chapter of [Inside Macintosh: PowerPC System Software](#)

Control Manager

The Control Manager provides facilities for drawing and processing user interaction with controls. New features for the Control Manager include:

- The Control Manager is now implemented in PowerPC code.
- There is a new edit text control variant supporting inline input.
- The edit text control now supports locking (disabling of user input).
- The static text control now supports text truncation.
- The icon control's icon can now be changed dynamically.
- Check boxes and radio buttons now support automatic toggling.
- Proportional scrolling is now available.
- Support for 32-bit control values has been added.
- Facilities for associating tagged data with controls (properties) have been added.
- A scrolling text box control (read-only text) has been added.
- A control region API has been added.
- A control validation API has been added.
- Control Manager now validates `ControlHandle` parameters and returns an error if they are invalid.
- Fonts used by controls can now be specified as a standard system font plus modifications.
- The clock control now returns more part codes.
- The edit text control now supports a validation callback procedure.

Related Materials:

- The [Appearance Manager](#), [Dialog Manager](#), [Menu Manager](#), and [Window Manager](#) sections
- Q&A [TB39](#), "Appearance Versions"
- The [High Level Toolbox](#) Technote Collection
- The [High Level Toolbox](#) Q&A Collection
- The [Control Manager Reference](#) chapter of *[Inside Macintosh: Mac OS 8 Toolbox Reference](#)*
- The [Control Manager](#) chapter of *[Inside Macintosh: Macintosh Toolbox Essentials](#)*
- *[Inside Macintosh: Macintosh OS 8 Human Interface Guidelines](#)*
- The [Appearance 1.0.3 SDK](#)

Device Manager

The Device Manager provides a programming interface for communications between applications and device drivers (typically code that communicates with particular hardware devices, although some device drivers do not actually drive devices).

- There are three new Driver Gestalt selectors allowing developers to return, among other things, color icons for their disk drives. Table 6 describes the new selectors.

Table 6. Driver Gestalt selectors introduced in Mac OS 8.5.

Selector Name	Value	Information Returned
kdgPhysDriveIconSuite	'dics'	A pointer to a IconFamily ('icons') data structure that can be used to represent the disk driver's physical drive (formerly in csCode 22) in driverGestaltResponse.
kdgMediaIconSuite	'mics'	A pointer to a IconFamily ('icons') data structure for representing the disk driver's media (formerly in csCode 21) in driverGestaltResponse.
kdgMediaName	'mnam'	A pointer to a Pascal string describing the disk driver (formerly in csCode 21) in driverGestaltResponse.

Related Materials:

- The [Icon Services, Driver Services Library](#), Inside the System File [General](#) sections
- The [Devices](#) Technote Collection
- The [Devices](#) Q&A Collection
- The [Device Manager](#) chapter of [Inside Macintosh: Devices](#)
- [Designing PCI Cards & Drivers](#)
- Minow, M. [The New Device Drivers: Memory Matters](#). [Develop](#) 24
- The [PCI Driver 2.0 SDK](#)

Dialog Manager

The Dialog Manager manages user interactions with dialogs. New features in the Dialog Manager include:

- The Dialog Manager is now implemented in PowerPC code.
- The Dialog Manager now allows you to specify dialog timeouts (automatic dismissal after an idle time).
- It is now possible for developers to specify the event mask used inside of `ModalDialog`.

Related Materials:

- The [Appearance Manager](#), [Control Manager](#), [Menu Manager](#), and [Window Manager](#) sections
- Q&A [TB39](#), "Appearance Versions"
- The [High Level Toolbox](#) Technote Collection
- The [High Level Toolbox](#) Q&A Collection
- The [Dialog Manager Reference](#) chapter of [Inside Macintosh: Mac OS 8 Toolbox Reference](#)
- The [Dialog Manager](#) chapter of [Inside Macintosh: Macintosh Toolbox Essentials](#)
- [Inside Macintosh: Macintosh OS 8 Human Interface Guidelines](#)
- Franke, N. [Multipane Dialogs](#). [Develop](#) 23
- Lavoie, M. & Johnson, B. [KON & BAL's Puzzle Page: AppendDITL Apoplexy](#). [Develop](#) 29
- The [Appearance 1.0.3 SDK](#)

Display Manager

The Display Manager manages changes in both the display settings and the arrangement of monitors attached to the computer.

- With some display cards, resolutions that were marked as non-preset resolutions by the display card were not showing up in the recommended list of resolutions in the Monitors & Sound control panel. The Monitors & Sound control panel was checking the non-preset bit, while the Display Manager was not. The Display Manager would strip the lower timing and Monitors & Sound would strip the higher (not preset) timing so neither would appear in the timing list used to build the recommended list. The Display Manager now correctly observes the non-preset bit and the recommended list is built correctly.
- In some cases, an unreadable character was appended to the end of some names returned by the Display Manager. This has been corrected.
- A problem in the Display Manager and Monitors & Sound control panel that could cause a crash when a multiscan monitor was attached to a PowerBook has been corrected. This problem would occur upon waking a PowerBook when the monitor was disconnected during sleep if the PowerBook was put to sleep with the Monitors & Sound control panel open.
- A problem on some machines where the display enabler would not load at startup if a smart display was plugged into the built-in monitor connector has been corrected.

Related Materials:

- Q&A [QD55](#), "How to Get the Monitor ID as Displayed in the Monitors & Sound Control Panel"
- Technote [TN1105](#), "AppleVision Displays"
- Esfahani, C. & Miller, K. [Graphical Truffles: Dynamic Display Dilemmas](#). [Develop](#) 26
- The [Display Manager 2.0.2 SDK](#)

Disk Initialization Package

The Disk Initialization Package supports the formatting of disks.

- `DILoad` was loading foreign file system resources into the current heap zone (usually, the current application's heap) that could cause a crash if the application quit before the resources were unloaded. These resources are now loaded into the system heap.
- File System Manager-based file systems that support bad block sparing are now called to spare blocks when a disk driver verify request fails. This feature was unavailable in Mac OS 8.0 and Mac OS 8.1.
- The initial value of the Format popup menu in the disk initialization dialog now indicates the current format of the disk.

Related Materials:

- The [File System Manager](#) and [File Manager](#) sections
- The Disk Initialization Manager section in Technote [TN1041](#), "Inside Macintosh: Files Errata"
- The [Disk Initialization Manager](#) chapter of *Inside Macintosh: Files*

Drag Manager

The Drag Manager supports Drag-and-Drop operations between windows and applications. New features for the Drag Manager include:

- `ShowDragHilite` now uses the highlight color chosen by the user in the Appearance control panel.
- A potential crashing problem that could occur with translucent drags has been corrected in this version of the Drag Manager. In previous versions of the Drag Manager, a pointer to an unlocked relocatable block (a handle's master pointer) was being used in `SetDragImage`.

Related Materials:

- Technote [TN1043](#), "On Drag Manager Additions (Release 1.1)"
- Technote [TN1085](#), "Using the Drag Manager to Interact with and Manipulate File System Entitles"
- Q&A [TB28](#), "Drag Manager and windowKind 20"
- The [Drag Manager SDK](#)

Driver Services Library

DriverServicesLib provides utility routines for native drivers ('ndrv's). The following improvements in virtual memory support have been added to the DriverServicesLib:

- PrepareMemoryForIO now uses LockMemoryForOutput instead of LockMemory when the IOPreparationOptions are kIOIsOutput, but not kIOIsInput.
- With Mac OS 8.5, GetPageInformation returns the kPageIsLockedResident (kPageIsLocked) PageStateInformation bit correctly. When Virtual Memory is on, kPageIsLockedResident indicates that a page has been locked with LockMemory, or LockMemoryForOutput. Before Mac OS 8.5, resident pages with the kPageIsInMemory (kPageIsResident) PageStateInformation bit set always had the kPageIsLockedResident bit set even when they were not locked.
- With Mac OS 8.5, GetPageInformation returns a new PageStateInformation bit, kPageIsHeldResident. When Virtual Memory is on, kPageIsHeldResident indicates that a page is held in physical memory (but not necessarily locked) with HoldMemory, LockMemory or LockMemoryForOutput.

Related Materials:

- The [InterfaceLib](#) and [Virtual Memory Manager](#) sections
- [Designing PCI Cards & Drivers](#)

Event Manager

The Event Manager manages the delivery of events to applications. Changes in the Event Manager include:

- Parts of the Event Manager are now PowerPC-native.
- Formerly, the event queue was limited to 20 events, and keystrokes from faster typists could be lost. The size of the event queue has been increased to 48 elements.

Related Materials:

- The [Event Manager Reference](#) chapter of [Inside Macintosh: Mac OS 8 Toolbox Reference](#)
- The [Event Manager](#) chapter of [Inside Macintosh: Toolbox Essentials](#)

File Manager

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

- PBXGetVolInfo was returning an incorrect value in ioVNmFls when ioVRefNum contained either a working directory id or volume number zero (the default directory) when the call referred to a FSM-based file system volume. As described in the File System Manager section, this has been corrected so the File Manager now passes the working directory id or zero through to FSM-based file systems, rather than converting the value to a real volume reference number first.
- Under some circumstances, the hard disk on some PowerBook models would not spin down when it was instructed to do so by the Control Strip module. This has been corrected.
- UnmountVol would sometimes return fBsyErr (File busy) errors when called for some server-mounted volumes because the File Manager was leaving some working directories open.

This has been corrected.

- The `pleaseCacheBit` (bit 4) in the `ioPosMode` field is now supported by the disk cache. Read/Write requests to HFS/HFS Plus volumes can now set the `pleaseCacheBit` in `ioPosMode` to request caching for larger blocks of data. File System Manager clients can also set this bit in the `cacheOptions` parameter to `UTCachedReadIP`, `UTCachedWriteIP`, `UTVolCachedReadIP`, and `UTVolCachedWriteIP` requests. Setting this bit does not guarantee a request will be cached; however, it does increase the probability of a request being cached.
- `PBGetFPos` now clears all 16 bits of `ioPosMode`.
- The routines `PBAllocate` and `PBAllocContig` have been modified so they set the value of `ioActCount` before they return even if an error occurs during their execution.
- The minimum size for the disk cache is now 128K.
- Previously, the colons occurring in file names were returned as-is, making such files unreachable using the Pascal string-based APIs. Colons are now converted to question marks (as are other non-representable characters) and the File ID is inserted in the name.
- Calls to `PBCatMove` on HFS Plus-formatted volumes were creating oversized catalog file thread records, leaking space in the catalog file. This has been corrected.
- A problem where `PBGetCatSearchSync` could return inconsistent results when searching HFS Plus volumes has been corrected.
- When the HFS Plus code was saving the Volume Header information, it was sign-extending the attributes field, causing the upper bits to be set when the software lock bit (bit 15) was set. This sign extension no longer occurs.
- `PBCatSearch` has been modified so it does not return `catChangedErr` as often as it did in previous system releases.
- Placing a file ID in the directory ID field of a `CInfoPBRec` and calling `PBGetCatInfo` now correctly returns `dirNFErr`. Under Mac OS 8.1, this would incorrectly produce information about a file as if it were a directory. Before 8.1, it would return `fnfErr`.
- Added a new call, `PBHTrashVolumeCaches`, that flushes a volume and invalidates that volume's caches in the File Manager and disk cache. Bit 2 of the `vmAttrib` field returned by `GetVolParams` is set if a volume supports this call.
- HFS Plus volumes store all dates in UTC. In Mac OS 8.1, those dates were converted to/from local time using the current time zone and daylight savings time settings, which would cause the dates to appear to change by one hour when switching to/from daylight savings time. In Mac OS 8.5, the dates are converted using the daylight savings time setting that would have been in effect as of the date being converted.
- In Mac OS 8.1, when a HFS or HFS Plus volume was extremely fragmented and an operation required the extents B-tree to grow beyond its maximum number of extents, the File Manager would return error -127 and leave extra bits set in the volume bitmap. (Disk First Aid would report this as a minor problem in the volume's allocation bitmap.) The File Manager has been corrected so that under these circumstances it will return a `dskFullErr` and it will not leave extra bits set in the volume bitmap.
- A problem where a crash could occur when saving a log in MacsBug has been corrected. The File Manager was attempting to jump to location zero.
- The File Manager was not properly invalidating some caches when a volume was unmounted. The most common symptoms were missing items in Finder windows when mounting/unmounting a series of floppies or disk images (such as installing software from

floppies), or a disk switch dialog appearing asking for a disk with a garbage or blank name. The File Manager now correctly invalidates a volume's caches when it is unmounted.

- A problem that could occur when mounting a CD-ROM has been corrected. In these cases, mounting a CD-ROM would sometimes lead to a dialog stating the volume needed to be repaired. Under certain conditions after a crash this could cause corruption of writable volumes when they were mounted.
- A `PBAllocContig` call requesting more space than was available contiguously would return the correct error but still allocate a smaller amount of space. `PBAllocContig` has been changed so that under these conditions it no longer allocates any space.
- The File Manager no longer writes to the first sector of a resource fork every time it is closed. Formerly, the first sector of a resource fork contained a copy of the 112-byte MFS directory entry intended for use by the Finder and disk repair utilities in older versions of Mac OS. No current disk repair utilities rely on this information, nor does the Finder. Removal of this extra read/write cycle speeds up Finder copies for files containing resource forks.
- A problem that could occur when booting from a HFS volume has been corrected. If aliases to shared library files located on a Mac OS Extended volume were present in the Extensions folder, then these aliases would be modified in such a way that they would fail to resolve every time the system was rebooted. In Mac OS 8.5, such aliases resolve correctly.
- A problem where `PBDelete` could return -127 and corrupt the volume bitmap when trying to delete a file on a HFS Plus volume has been corrected. This problem would occur when a filename was originally converted to Unicode using the wrong text encoding (e.g., an application creating a file using a Japanese filename without the language kit installed, but later attempting to delete the file with the language kit installed).
- Fixed a problem where foreign (external) file systems could sometimes be called with asynchronous requests while interrupts were partially disabled. While this rarely occurred, when it did, it caused problems for some foreign file systems (such as AppleShare) that use Open Transport to communicate with file servers. All asynchronous requests to foreign file systems now start at deferred task time or at system task time, with interrupts fully enabled.

WARNING:

Copying files near the 2 GB HFS file size limit to an HFS volume will fail when the destination file's physical size goes beyond the 2 GB size limit due to larger allocation block sizes being used on the destination volume.

Related Materials:

- The [Disk Initialization Package](#) and [File System Manager](#) sections
- Technote [FL16](#), "File Manager Performance and Caching"
- The File Manager section in Technote [TN1041](#), "Files Errata"
- The [Files](#) Technote Collection
- The [Files](#) Q&A Collection
- The [File Manager](#) chapter of [Inside Macintosh: Files](#)

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.

- `XGetVolInfo` requests (like `GetVolInfo` requests) to a FSM-based file system are now passed a working directory number or the default volume number (the value zero) in `ioVRefNum` if the File Manager caller used a working directory number or the default volume number. Under Mac OS 8.1, `ioVRefNum` would always contain the real volume reference number when passed to a FSM-based file system. This change lets the FSM-based file system determine what to return in `ioVNmFls` correctly.
- `UTTrashBlocks` now correctly returns an `OSErr` result (either `noErr`, `rfNumErr`, or `fnOpnErr`). Before Mac OS 8.5, `UTTrashBlocks` returned the file reference number passed in as the `fileRefNum` parameter.
- The File Manager's disk cache now range-checks the buffer parameter passed to `UTReleaseBlock` and `UTMarkDirty`. If the buffer parameter is bad, those calls will no longer crash the system.

Related Materials:

- The [File Manager](#) section
- The [File System Manager 1.2 SDK](#)

Folder Manager

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

- Folder Manager defines new auto-routings for preferences files of type `'pref'` (the Preferences folder), shared libraries (the Extensions folder), data fork fonts (the Fonts folder), Open Font Architecture plug-ins (the Extensions folder), and AppleTalk extensions (the Extensions folder).
- `FindFolder` now returns the actual location of the Printer Descriptions folder.
- Auto-routing has been added for files of type `'issp'` (to Internet Search Sites).
- Files of type `'JPEG'` are now auto-routed to the Desktop Pictures folder.
- As in pre-Mac OS 8.0 versions of the system software, the `FindFolder` folder cache now detects volume changes and keeps the cache up to date. In Mac OS 8.0 and Mac OS 8.1, it was possible for `FindFolder` to return stale data.
- Several new folders have been defined for Mac OS 8.5. Table 7 lists these new folders and their types.

Table 7. New folders for Mac OS 8.5.

Folder Name	OSType	Description
Launcher Items	'laun'	Items appearing in the Launcher control panel. Items included in folders with names beginning with a bullet (option-8) character will appear as a separate panel in the Launcher window.

Internet Search Sites	'issf'	Internet search site specification files used by the Find application when it accesses Internet search sites. Files of type 'issp' are auto-routed to this folder.
TheFindByContent Folder	'fbcf'	An invisible folder located in a volume's root directory. This folder is used to store files created by 'Find by Content'.
Find	'fnds'	Contains files used by the Find facilities in Mac OS 8.5. This folder is located in the Extensions folder.
Installer Logs	'ilgf'	A location for saving installer log files.
ColorSync Profiles	'prof'	A location for storing ColorSync profiles.
Appearance	'appr'	Appearance-related materials
Theme Files	'thme'	The location for storing theme files
Sound Sets	'snds'	The location for storing appearance-related sound sets
Desktop Pictures	'dtpf'	The location for storing desktop picture files. Files of type 'JPEG' are auto-routed into this folder when dropped into the System Folder.
Favorites	'favs'	The location for storing Internet location files, aliases, and aliases to other frequently used items. Facilities for adding items into this folder are found in Contextual Menus, the Finder, and Navigation Services, et al.
Scripts	'scr ^f '	A location for saving AppleScript scripts.
Recent Applications	'rapp'	Apple Menu Items saves aliases to recent applications here.
Recent Documents	'rdoc'	Apple Menu Items saves aliases to recently opened documents here.
Recent Servers	'rsvr'	Apple Menu Items saves aliases to recently mounted servers here.
Speakable Items	'spki'	A location for scripts and items recognized by speech recognition.
Internet	'int ^f '	A location for saving Internet-related applications, resources, and tools.
Location Manager Modules	'walk'	Location manager modules are stored here.
Location Manager Prefs	'trip'	Location manager preferences files are saved here.
Locations	'fall'	Files containing configuration information for different locations are stored here.

Related Materials:

- The [Finder: Autorouting and Extended Routing](#) section
- The [Finder Interface Reference](#) chapter of *Inside Macintosh: Mac OS 8 Toolbox Reference*
- The [FindFolder](#) description in the [Finder Interface](#) chapter of *Inside Macintosh: Macintosh Toolbox Essentials*
- The [Using the System Folder and Its Related Directories](#) section in the [Finder Interface](#) chapter of *Inside Macintosh: Macintosh Toolbox Essentials*
- The [File Manager](#) chapter of *Inside Macintosh: Files*

Font Manager

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

- The Font Manager is now PowerPC-native.
- A problem where an italic bitmap font would incorrectly override a TrueType font of the same size displaying letters with the wrong style has been corrected. Now, the TrueType font will be displayed unless the style of the bitmap font matches exactly.
- Font Manager no longer allocates or de-allocates caches on a per-processes basis. A system-wide font cache is allocated at system startup in the system heap. All processes share this same cache.
- The Font Manager does more parameter checking than in previous versions.
- Font Manager now purges fonts from the font cache more aggressively.
- All fonts shipping with Mac OS 8.5 (except VT102) include the new Euro glyph.
- The US version of Mac OS 8.5 ships with the following new fonts: Capitals, Gadget, Sand, Techno, and Textile.

Related Materials:

- The [ATSUI](#) and [QuickDraw Text](#) sections
- The [Text](#) Technote Collection
- The [Font Manager](#) chapter of *Inside Macintosh: Text*
- Technote [1140](#): "The Euro Currency Symbol"

Help Manager

The Help Manager is responsible for the drawing and display of Help Balloons and the Help menu.

- The Help Viewer application can now be accessed directly from the Help Menu.

Related Materials:

- The [Apple Help](#) section
- The [Help Systems](#) Technote Collection
- The [Help Systems](#) Q&A Collection
- The [Adding Menu Items to the Help Menu](#) section in the [Help Manager](#) chapter of [Inside Macintosh: More Macintosh Toolbox](#)

Icon Services

New for Mac OS 8.5, 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).

- Icon Utilities is now implemented in PowerPC code. (Icon Services is also native.)
- Icon Services is Appearance Manager-aware, and will return information consistent with the active appearance as appropriate. (If your program retrieves icons from the System file via Icon Utilities or other means, these icons will not differ according to appearance.)
- To avoid duplication of data and for better performance, Icon Services maintains a system-wide icon cache shared by all applications. Icon data returned is common to all applications, so it is possible for two different applications to obtain an icon reference that refers to the same icon data. A reference count is maintained for icons loaded into the cache, and icons are removed from the cache only if the reference count falls to zero.
- The new 'icons' resource contains data for 4 icon sizes (mini, small, large, huge) at 4 color depths (1-bit, 4-bit, 8-bit, and 32-bit) and two kind of masks (1-bit masks and 8-bit masks). Deep (8 bit) masks allow for the specification of transparent drawing effects. Each pixel in a deep mask indicates the level of transparency desired for drawing a particular pixel. Transparency levels range from 0 (transparent) through 255 (opaque).
- Prior to Mac OS 8.5, Icon Utilities allowed only 34 arbitrary colors to highlight properly (they were known as the Apple icon colors). This limitation has been removed.
- Drivers can now provide color icons by implementing the `kdgMediaIconSuite` Driver Gestalt selector as described in the [Device Manager](#) section.

Related Materials:

- The [Finder: New Finder Display Facilities](#) and [Device Manager](#) sections
- [Inside Macintosh: Icon Services \(preliminary draft\)](#)
- Q&A [TB35](#), "Obtaining Standard Icons"
- The [Icon Utilities](#) chapter of [Inside Macintosh: More Macintosh Toolbox](#)

InterfaceLib

InterfaceLib provides a Code Fragment Manager interface to many of the Mac OS routines on PowerPC-based computers. The following additions and changes are present in this most recent version of the InterfaceLib.

- The low-memory accessor `LMGetABusVars` is now deprecated and should not be used, as it behaves differently on 68K and PowerPC computers. The new accessor `LMGetABusGlobals` that retrieves the long at `0x02D8` should be used in its place. `LMGetABusVars` will continue to return the value `0x02D8`. The following new accessors were added: `LMGetABusGlobals`, `LMSetABusGlobals`, `LMGetABusDCE`, and `LMSetABusDCE`.
- The following low-memory accessors have been added. They are useful for determining how long it has been since any user activity has occurred.
 - `LMGetKeyTime` -- the value returned by `TickCount` when the last keystroke was received.
 - `LMGetMBTicks` -- the value returned by `TickCount` when the mouse button was last pressed.

Formerly, these values could be accessed using the routines shown in Listing 2.

Listing 2. Low-memory accessors for `KeyTime` and `MBTicks`.

```
long LMGetMBTicks (void)
{
    return (*(long *)0x016E);
}

long LMGetKeyTime (void)
{
    return (*(long *)0x0186);
}
```

Also, routines for setting these values have also been added: `LMSetKeyTime` and `LMSetMBTicks`.

- `AddDrive`

There was a problem in the parameter handling of the PowerPC `AddDrive` glue code in `InterfaceLib` that would mangle the drive number before it was called through to the 68K code. This problem has been corrected.

- New [Alias Manager](#) routines

`IsAliasFile`, `ResolveAliasWithMountFlags`, and `ResolveAliasFileWithMountFlags` are now included in `InterfaceLib`.

- New routines added to `InterfaceLib` that were previously only available from 68K code:

`DriverInstall`, `DriverInstallReserveMem`, `DIXFormat`, `DIXZero`, `DIREformat`, `PBUnmountVolImmed`, `ReallocateHandleSys`, `LockMemoryForOutput`, `MakeMemoryResident`, `ReleaseMemoryData`, `MakeMemoryNonResident`, `FlushMemory`, `InstallExtensionNotificationProc`, `RemoveExtensionNotificationProc`, `InstallExtensionTableHandlerProc`, `RemoveExtensionTableHandlerProc`, `FlushCodeCacheRange`, `NewCServiceWindow`, `UCTextServiceEvent`, and `ataManager`

- The rest of the `PurgeSpace` calls that were previously only available from 68K code have been added to `InterfaceLib`:

`PurgeSpaceTotal`, `PurgeSpaceContiguous`, `PurgeSpaceSysTotal`, and `PurgeSpaceSysContiguous`

- All the File System Manager calls have been added to `InterfaceLib`:

`UTAllocateFCB`, `UTReleaseFCB`, `UTLocateFCB`, `UTLocateNextFCB`, `UTIndexFCB`, `UTResolveFCB`, `UTAllocateVCB`, `UTAddNewVCB`, `UTDisposeVCB`, `UTLocateVCBByRefNum`,

UTLocateVCBByName, UTLocateNextVCB, UTAlocateWDCB, UTReleaseWDCB, UTResolveWDCB, UTFindDrive, UTAdjustEOF, UTSetDefaultVol, UTGetDefaultVol, UTEjectVol, UTCheckWDRNum, UTCheckFileRefNum, UTCheckVolRefNum, UTCheckPermission, UTCheckVolOffline, UTCheckVolModifiable, UTCheckFileModifiable, UTCheckDirBusy, UTParsePathname, UTGetPathComponentName, UTDetermineVol, UTGetBlock, UTReleaseBlock, UTFlushCache, UTMarksDirty, UTTrashVolBlocks, UTTrashFileBlocks, UTTrashBlocks, UTCacheReadIP, UTCacheWriteIP, UTBlockInFQHashP, UTVolCacheReadIP, UTVolCacheWriteIP, InstallFS, RemoveFS, SetFSInfo, GetFSInfo, InformFSM, and InformFFS

- The following [DriverServicesLib](#) calls have been added to InterfaceLib:

BlockZero, BlockZeroUncached, IncrementAtomic, DecrementAtomic, AddAtomic, BitAndAtomic, BitOrAtomic, BitXorAtomic, CompareAndSwap, IncrementAtomic8, DecrementAtomic8, AddAtomic8, BitAndAtomic8, BitOrAtomic8, BitXorAtomic8, IncrementAtomic16, DecrementAtomic16, AddAtomic16, BitAndAtomic16, BitOrAtomic16, BitXorAtomic16, TestAndSet, and TestAndClear.

Previously these routines were documented as only available to native drivers ('ndrv's). The InterfaceLib versions of these routines can be called by other types of code.

COMPATIBILITY NOTE

Developers linking against newer versions of InterfaceLib who would like to have their products run with previous versions of the system software should weak-link against the new symbols. Unless this is done, the Code Fragment Manager will refuse to launch applications using the new symbols when an older version of InterfaceLib is being used. Developers weak-linking against any of the new symbols in InterfaceLib should check to ensure that the routines they weak-link against are defined before calling them.

Related Materials:

- The [DriverServicesLib](#) and [Alias Manager](#) sections
- Technote [TN1083](#), "Weak-Linking to a CFM-based Shared Library"
- Technote [TN1127](#), "In Search of Missing Links"

List Manager

The List Manager provides a standard user interface for drawing and managing user interaction with lists of items.

- The List Manager is now implemented in PowerPC code.
- Support for live scrollbar indicator dragging has been added.
- The proportional scrollbar indicator is now supported.
- When deleting or adding cells to a list while drawing is turned off -- by calls to `LSetDrawingMode` (a.k.a. `LDoDraw`) -- in certain cases it was possible for the `ListHandle`'s bounds to become slightly scrambled and draw incorrectly when drawing was turned back on. This problem has been present in the List Manager since its introduction, and it has been corrected in Mac OS 8.5.
- List click loops can now be written in PowerPC-native code using the Mixed Mode glue available in "Lists.h".

WARNING:

Writing list click loops in PowerPC-native code is not backward-compatible with system releases prior to Mac OS 8.5. For compatibility with all systems prior to Mac OS 8.5, you must still write click loops using 68K assembly or a 68K assembly stub.

Related Materials:

- Technote [TB13](#), "'LDEF' Madness"
- Q&A [TB09](#), "List Manager & LClick"
- The [List Manager](#) chapter of [Inside Macintosh: More Macintosh Toolbox](#)

MathLib

MathLib provides a software implementation of floating point support routines for PowerPC computers. All PowerPC processors contain built-in floating point processing capabilities. The routines provided in MathLib supplement those facilities.

- MathLib is now present in the System file's data fork, and as such it is eligible for paging.

Related Materials:

- [Inside Macintosh: PowerPC Numerics](#)

Memory Manager

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

- Under some circumstances, the RAM Disk could not be turned off. This has been corrected.
- The Memory Manager is now present in the System file's data fork, and as such it is eligible for paging.

Related Materials:

- The [Virtual Memory Manager](#) section
- Technote [IM_ERRATA 02](#), "Inside Macintosh -- Memory Errata"
- The [Memory](#) Technote Collection
- The [Memory](#) Q&A Collection
- The [Memory Manager](#) chapter of [Inside Macintosh: Memory](#)

Menu Manager

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

- Most of the Menu Manager is now implemented in PowerPC code.
- Support for associating tagged data with menus and menu items (properties) has been added.
- Menu contents for menus that use the system 'MDEF' are now cached even when is VM on.
- It is now possible to remove the current application's name from the application menu.
- Support for disabling icons appearing in menus has been added.
- The menu bar is now drawn in condensed-style text, and long menu titles are truncated if the screen is too narrow to display the entire menu bar contents.
- Support for excluding the menu mark column has been added.
- Support for more than 255 submenus has been added. `SetMenuItemHierarchialID` can be used to set the full range of menu IDs (-32767 to 32767).
- Support for enabling and disabling menu items at indexes greater than 31 has been added.
- Sorting in `AppendResMenu` was not working as expected in non-English languages. `RelString` was being called instead of `CompareString`. `AppendResMenu` now uses `CompareString` so sorting now works as expected with non-English languages.
- Support for showing and hiding the menu bar has been added.
- New append/insert APIs without meta-character interpretation have been added.
- The system 'MDEF' now supports all keyboard modifiers and glyphs.
- Support for setting the font used to draw a menu has been added. Developers should use this new API rather than setting the low-memory globals `SysFontFam` and `SysFontSize`.

COMPATIBILITY NOTE

In Mac OS 8.5, the Menu Manager does not attempt to modify the behavior of third-party 'MDEF's for correct colorization. Application vendors including custom 'MDEF's with their products must revise their products to support Appearance.

COMPATIBILITY NOTE

The Menu Manager does not modify the behavior of the Event Manager to force `Button` to return `true` while tracking sticky menus. 'MDEF' authors who depend on `Button` returning `false` values must modify their products.

Related Materials:

- The [Appearance Manager](#), [Control Manager](#), [Dialog Manager](#), and [Window Manager](#) sections
- Q&A [TB39](#), "Appearance Versions"
- The [High Level Toolbox](#) Technote Collection
- The [High Level Toolbox](#) Q&A Collection
- The [Menu Manager Reference](#) chapter of *Inside Macintosh: Mac OS 8 Toolbox Reference*
- The [Menu Manager](#) chapter of *Inside Macintosh: Macintosh Toolbox Essentials*
- *Inside Macintosh: Macintosh OS 8 Human Interface Guidelines*
- The [Appearance 1.0.3 SDK](#)

Navigation Services 1.1

Navigation Services provides new Open and Save dialogs, allowing users to locate and select files and other resources. Navigation Services 1.0 is delivered as a separate shared library in the Navigation Services SDK. In Mac OS 8.5, Navigation Services 1.1 is built in to the System file.

- Event procedures now receive appropriate `mouseDown` events before they are processed by Navigation Services.
- Event procedures now receive a slightly different parameter block (which is backward-compatible with 1.0 callers) that allows them to see the dialog item index of an item that has been selected by the user.

COMPATIBILITY NOTE

Navigation Services effectively obsoletes the Standard File package. Developers interested in future Mac OS compatibility are encouraged to replace their calls to Standard File with calls to Navigation Services.

Related Materials:

- The [Standard File](#) section
- Q&A [TB38](#), "Navigation Services Versions"
- Q&A [TB40](#), "Navigation Services and memFullErr"
- The [Navigation Services SDK](#)

Notification Manager

The Notification Manager provides a mechanism for programs which are not the frontmost application to notify the user of specific events.

- The Notification Manager was calling `GetResource` when the interrupt level mask was set to 7 (even though it wasn't executing at interrupt time). This has been corrected.

Related Materials:

- Technote [PS505](#), "Notification Manager Q&As"
- Technote [TN1026](#), "The Notification Manager: Problems & Fixes"
- The [Notification Manager](#) chapter in *Inside Macintosh: Processes*

PC Card Manager

The PC Card Manager handles software issues related to PC Cards.

- If a PC Card was ejected during sleep (using the pin hole method) and replaced with a different card, the new card would not be recognized after sleep and the old card icon would remain on the desktop. The PC Card Manager now registers the new card correctly when a card is replaced in this way.
- The PC Card Manager has been changed to accommodate PC Cards that do not contain a CIS (Card Information Structure -- information normally stored on a card which describes the card). Prior to this change, such a card would simply be ejected. Now, the PC Card Manager uses other data, such as the PCI configuration registers, to establish information about such cards.

Related Materials:

- Technote [TN1099](#), "Power Management & PC Card Manager 3.0"
- The [PC Card Manager 2.0 and 3.0 SDK](#)

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 an AppleTalk network.

- The PPC Browser now handles low memory situations without crashing.
- In previous versions of the PPC Browser, the NBP lookup for the list of computers was using too short of a delay. As a result, the list of computers would flicker over slower network connections. The NBP lookup now uses a longer delay so the list is drawn with less flicker.
- The PPC Browser now performs extensive internal error checking.

Related Materials:

- Q&A [IC02](#), "noResponseErr from PPC Toolbox"
- Q&A [IC01](#), "Closing PPC ports"
- Technote [IC515](#), "PPC Toolbox Q&As"
- The [Program-to-Program Communications Toolbox](#) chapter in [Inside Macintosh: Interapplication Communication](#)

Process Manager

The Process Manager shares the processor among multiple applications. Mac OS 8.5 includes the following new features in the Process Manager:

- Under some circumstances, the cursor would not change to the cross hairs as expected for small area screen captures. This has been corrected.
- `LaunchApplication` now honors alias files.
- Process Manager is slightly more strict about when it will call `jGNEFilter` for "fake" `keyDown` events created by the Text Services Manager, eliminating some duplicate events.
- During application switches, the Process Manager now checks if a `mousedown` was pending in the event queue for the foreground application, or any floating windows in the Text Services Manager layer. This correction eliminates the possibility of a mouse click in one of the foreground application's windows sneaking through to the application being switched in.
- The Process Manager will no longer launch invisible 'appe' files.
- If a process unmounted a volume that was the default volume for one or more other processes, the default volume was not correctly updated for those other processes when they were switched in. This could result in a File Manager disk switch dialog appearing asking for a disk with a garbage or blank volume name. The Process Manager now ensures that the File Manager's default volume is valid when switching processes.

Related Materials:

- Technote [TN1070](#), "Background-Only Applications"
- Technote [PS510](#), "Process Manager Q&As"
- The [Processes](#) Q&A Collection
- The [Process Manager](#) chapter in [Inside Macintosh:Processes](#)

QuickDraw

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

- Most of QuickDraw is now PowerPC-native (with the exception of the Palette Manager and the Picture Utilities).
- Previously, QuickDraw pictures containing invalid 0x9A (direct 24-bit images) opcodes recorded on 68K machines would not play back correctly on PowerPC machines. This has been corrected.
- A problem where the 16-bit encoding routines for QuickDraw pictures would sometimes produce invalid data has been corrected.
- A problem where larger font sizes would not draw has been corrected. Fonts are now drawn at their largest possible size if a size larger than the maximum font display size is requested.
- The Roman Script Utilities (`DrawJustified`, `CharToPixel`, etc.) have been moved into QuickDraw Text.
- 'sbit' support, currently available only via a WorldScript II patch, has been rolled into the standard system. This allows systems using two-byte characters to be booted with extensions turned off.
- All `CopyDeepMask` transfer modes now work correctly.
- QuickDraw now performs extensive parameter checking. For parameter errors, `QDError` will return the `paramErr` result code. In many cases, `QDError` returns errors where it did not before.
- A problem where passing `bkPixPat` either to `FillCRgn` or `FillCRect` would corrupt the current graphics port has been corrected. It is now possible to provide the current background `pixpat` as a parameter to either the `FillCRgn` or the `FillCRect` routine.

WARNING:

Calling either

```
FillCRect(&myRectangle,
          ((CGrafPtr) qd.thePort)->bkPixPat);
```

...or...

```
FillCRgn(myRegion,
          ((CGrafPtr) qd.thePort)->bkPixPat);
```

...in any version of color QuickDraw prior Mac OS 8.5 will destroy the current `bkPixPat` in the `GrafPort`.

- When recording pictures, QuickDraw would read frame buffers for raster images one byte at a time. This produced some compatibility problems when reading frame buffer data from some non-Apple devices. For greater compatibility, frame buffers are now read in pixel size chunks that are buffered internally before being passed to the RLE encoding routines.
- Nearly all of QuickDraw is now present in the system file's data fork, and as such it is eligible for paging. The cursor management parts of QuickDraw are not eligible for paging and remain in memory at all times.

COMPATIBILITY NOTE

The new QuickDraw does more parameter checking than all previous versions of QuickDraw. As a result, developers will find the new QuickDraw will facilitate their debugging efforts by bringing to light problems that may not have been so obvious before. It is strongly suggested that developers verify their products against the new QuickDraw to ensure they are calling QuickDraw correctly.

- Related Materials:
 - The [QuickDraw Text](#) section
 - The [QuickDraw](#) Technote Collection
 - The [QuickDraw](#) Q&A Collection
 - [*Inside Macintosh: Imaging with QuickDraw*](#)

QuickDraw Text

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

- A new Gestalt selector, `gestaltQDTextVersion ('qdtx')`, has been defined in this release of QuickDraw Text. With this release of QuickDraw Text, the value returned is `gestaltAllegroQDText`.

COMPATIBILITY NOTE

The new QuickDraw Text routines described in this section are available only if the result returned by a call to `Gestalt` with the selector `gestaltQDTextVersion` is greater than or equal to the constant `gestaltAllegroQDText`.

- Support for anti-aliased text has been added to QuickDraw Text. Anti-aliased text drawing will occur if it is enabled, the destination device pixel depth is greater than or equal to 8 bits, there is sufficient memory available, the requested font size for drawing is larger than the minimum size for anti-aliased drawing, and the font being drawn is an outline font. Two new routines for accessing the state of anti-aliased text drawing can be called from PowerPC applications linking against the shared library `FontManager` included in the System file. The routines are defined as follows:

IsAntiAliasedTextEnabled

```
Boolean IsAntiAliasedTextEnabled(SInt16* outMinFontSize);
```

`outMinFontSize` is an optional parameter and may be set to `NULL`. If `outMinFontSize` is not `NULL`, then the sixteen bit integer at that address will be set to the current lower size limit for antialiasing text (as set in the Appearance Control Panel).

The function `IsAntiAliasedTextEnabled` returns `true` if anti-aliased text display is enabled.

SetAntiAliasedTextEnabled

```
OSStatus SetAntiAliasedTextEnabled(Boolean inEnable,
                                     SInt16 inMinFontSize);
```

`inEnable` indicates the desired state of the anti-aliased text drawing. If `inEnable` is `true`, then both anti-aliased text drawing will be turned on and the QuickDraw Text global variable `outlinePreferred` will be turned on.

`inMinFontSize` defines the smallest size font that should be drawn as anti-aliased text. This value must be in the range 1 through 128. If the `inMinFontSize` parameter is outside of this range, then the default value 12 is used.

`SetAntiAliasedTextEnabled` always returns `noErr`.

NOTE: For fonts which have both bitmaps ('`NFNT`'s) and outlines ('`sfnt`'s), the implicit change in `outlinePreferred` may change the metrics for the bitmap font sizes.

WARNING:

Calls to `SetAntiAliasedTextEnabled` should only be made under special circumstances when greater control over text display during particular operations is desired. Developers changing the state of anti-aliased text drawing must restore its original state before returning control to the system. Failure to do so may result in other parts of the system, most notably the Appearance control panel, becoming out of sync with the actual state of anti-aliased text drawing.

- Two new routines for retrieving the pixel dimensions of text that will be drawn on the screen have been added to QuickDraw Text. These routines can be called from PowerPC applications linking against the shared library `FontManager` in the System file. The new routines are defined as follows:

QDTextBounds

```
void QDTextBounds(short byteCount,
                  const void* textAddr,
                  Rect* bounds);
```

`byteCount` is the number of bytes of text contained in the buffer located at the address contained in the value `textAddr`.

`textAddr` points to `byteCount` bytes of textual data.

`bounds` is a pointer to a `Rect` structure whose coordinates will be calculated by `QDTextBounds`. On return, `bounds` will contain the bounding coordinates for the entire image (including parts of the image that may extend beyond the first and last pen positions after the text has been drawn) that will be drawn for the text (given the font settings in the current `GrafPort`). Note that the coordinates returned are relative to the current pen position (as if the pen were located at the origin). For example, in Listing 3, we use `QDTextBounds` to draw a rectangle around the text's image. Notice how in this example the bounds returned by `QDTextBounds` are offset using the current pen position, so that the text drawing and the rectangle drawing occur using the same coordinate system.

Listing 3. Using `QDTextBounds` to discover where text will be drawn.

```
Rect bounds;
char *text = "sample text";
Point where;
SetPt(&where, 100, 100);
MoveTo(where.h, where.v);
PenSize(1,1);
QDTextBounds(text, strlen(text), &bounds);
OffsetRect(&bounds, where.h, where.v);
InsetRect(&bounds, -1, -1);
FrameRect(&bounds);
DrawText(text, 0, strlen(text));
```

The leftmost edge of the text's image can either be to the right or the left of the pen position, and the rightmost edge of the text's image may be to the left or the right of the final pen position.

FetchFontInfo

```
OSErr FetchFontInfo(SInt16 fontID,
                    SInt16 fontSize,
                    SInt16 fontStyle,
                    FontInfo* info);
```

`fontID` is the font ID number for a font.

`fontSize` is the font size in pixels.

`fontStyle` contains the font style flags for the font.

`info` is a pointer to a font information record where the result will be stored.

`FetchFontInfo` returns the same information as `GetFontInfo`, except, rather than gathering information about the font settings from the current `GrafPort`, this information is provided as parameters to the routine. If `FetchFontInfo` returns an error, the fields in the `FontInfo` record will be set to zero. (The error code returned is the value returned by `FMSwapFont`.)

Related Materials:

- o The [ATSUI](#), [Font Manager](#), [QuickDraw](#), and [Appearance](#) control panel sections
- o Technote [TE20](#), "The Appearance of Text"
- o The [QuickDraw Text](#) chapter of *Inside Macintosh:Text*

Script Manager

The Script Manager provides functions for controlling, modifying, and accessing the features of both Roman and non-Roman script systems.

- Simple scripts (scripts not requiring the Worldscript extension) were not being loaded when the system was booted with extensions turned off. This has been changed; now, simple scripts are loaded when extensions are turned off.
- The keyboard menu bar icon was being drawn twice whenever a new keyboard layout was selected. This has been fixed.
- WorldScript Power Adapter is not loaded if it is installed by a language kit or other installer, since Mac OS 8.5 has this functionality built into the System file. WorldScript II is no longer loaded differently than other extensions.
- Support for Unicode scripts and Unicode keyboard layouts has been added.
- Font and keyboard synchronization can now be turned off. This can be done either in the Keyboard control panel or by setting the `smfDisableKeyScriptSync` flag in the `smGenFlags` Script Manager variable. Calls to `KeyScript` check this flag to determine if the current keyboard script should be synchronized with the current script. To force keyboard script synchronization for individual calls to `KeyScript` while keyboard script synchronization is turned off, developers can set bit 7 (`smKeyForceKeyScriptBit`) in the code argument passed to `KeyScript`. Calls to `KeyScript` will not synchronize the current keyboard script if the `smfDisableKeyScriptSync` flag in the Script Manager variable `smGenFlags` is set, unless the `smKeyForceKeyScriptBit` bit is set in the `KeyScript` call's code parameter.

Related Materials:

- The [Script Manager](#) chapter of *[Inside Macintosh:Text](#)*

Serial Driver

The Serial Driver provides software support for serial communications hardware using the Device Manager API. The Mac OS 8.5 Serial Driver includes the following change:

- A problem where some machines would not reboot without the "Serial (Built-in)" extension in the Extensions folder has been corrected.

Related Materials:

- Q&A [DV30](#), "Serial (Built-In)"
- Technote [TN1018](#), "Understanding the SerialDMA Driver"
- Technote [TN1119](#), "Serial Port Apocrypha"
- The [Serial Driver](#) chapter of *[Inside Macintosh: Devices](#)*

Sound Manager

The Sound Manager provides facilities for playback and recording of digitized sounds.

- The `siActiveChannels SPBSetDeviceInfo` call was not being accepted on 6100, 7100, or 8100 PowerMacs. This has been corrected.
- It is now possible to change recording quality while recording.
- Chained calls to `SPBRecord` now work when the recording buffer size is smaller than the hardware interrupt buffer size.
- Sound input source settings made in Monitors & Sound are now remembered across boots on 6100, 7100, and 8100 machines.
- `siRecordingQuality` now works as expected.
- The Power Mac 4400/200 always reported that the microphone is not connected regardless of whether it actually was. This problem has been corrected.

Related Materials:

- Technote [TN1048](#), "Some Sound Advice: Getting the Most Out of the Sound Manager"
- Technote [TN1124](#), "New Sound Input Driver Features"
- Technote [TN1108](#), "Unknown Sound Features"
- The [Sound Manager](#) chapter of *Inside Macintosh: Sound*
- Olson, K. [Sound Secrets](#). [Develop](#) 24

Standard File

Standard File provides a consistent user interface for accessing files by way of the Open and Save As dialog boxes.

- The number of file types that can be specified for Standard File's `SFGetFile`, `SFPGetFile`, `StandardGetFile`, and `CustomGetFile` routines has been increased from 32 types to 256 types.
- A problem where the arrow navigation keys were not working as expected on some PowerBook models has been corrected.
- Aliases to special folders (for example, the Fonts folder) no longer show up in Standard File as applications.

COMPATIBILITY NOTE

Navigation Services effectively obsoletes the Standard File package. Developers interested in future Mac OS compatibility are encouraged to replace their calls to Standard File with calls to Navigation Services.

Related Materials:

- The [Navigation Services](#) section
- The [Standard File Package](#) chapter of *Inside Macintosh: Files*
- The [Navigation Services SDK](#)

Text Services Manager 1.5

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, etc.

- The Text Services Manager (TSM) has removed the restriction of input methods and text services to Japanese, Korean, and Chinese (Traditional and Simplified) scripts, and now supports input methods in all scripts such as Roman, Indic, and Thai.
- Keyboards and input methods can now generate Unicode directly. Applications that support TSM and create Unicode TSM Documents with the new interface type `kUnicodeDocument` can receive all input as arbitrary-length Unicode text directly via Apple events. Unicode text in TSM Apple events is always contained in a descriptor with `AEKeyword keyAETheData`, `DescType typeUnicodeText`. Apple events with `AEEventID kUpdateActiveInputArea` continue to be used for input processed by an Input Method. The new `AEEventID kUnicodeNotFromInputMethod` is used for any other type of input, such as when the input source is a keyboard layout (Unicode or otherwise) or when an input method does not handle a key event. This Apple event also contains the original low-level key event record as `AEKeyword keyAETSMEventRecord` and `DescType typeLowLevelEventRecord`, so that most key event processing formerly done in an application's event loop can now be performed in the application's Apple event handlers.
- For compatibility, text is converted between Unicode and Mac OS encodings as necessary. Text from Unicode input sources is automatically converted to Mac encodings for delivery to applications that don't use Unicode TSM Documents; text from Mac OS encoding input sources is converted to Unicode for delivery to applications using Unicode TSM Documents. Similarly, application text requested by an input method (with the `kGetSelectedText` Apple event) is converted as necessary.
- TSM calls the new Input Method API `UCTextServiceEvent` instead of `TextServiceEvent` when when sending input text to a Unicode input method.
- TSM has absorbed functionality previously available as TSM trap patches, and itself no longer installs patches.
- TSM has been modified to support optional font and keyboard synchronization. If the `smfDisableKeyScriptSync` flag is set in the `smGenFlags` Script Manager variable, calls to `ActivateTSMDocument` will not synchronize the current keyboard script with the internal keyboard script state of the `TSMDocument`. The `ActivateTSMDocument` call is the only TSM call that calls through to `KeyScript`. For more information about optional font and keyboard synchronization, see the [Script Manager](#) section.

Related Materials:

- The [ATSUI](#), [Script Manager](#), and [Keyboard](#) control panel sections
- Technote [TE27](#), "Inline Input for TextEdit with TSMTE"
- The [Text Services Manager](#) chapter of *Inside Macintosh: Text*
- Griffith, T. [Gearing Up for Asia With the Text Services Manager and TSMTE](#). Develop 29

Time Manager

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

- The Time Manager will now return an error if `PrimeTime` is called after `RmvTime` was called.

COMPATIBILITY NOTE

The current C and Pascal Time Manager APIs do not return the result from `InsTime`, `InsXTime`, `PrimeTime`, and `RmvTime`. The result from Time Manager functions is returned in the 68K register `D0`. Simple 68K inline or PowerPC Mixed Mode glue code can be used to call those Time Manager functions and get their results.

- Related Materials:
 - Technote [PS520](#), "Time Manager Q&As"
 - The [Time Manager](#) chapter in [Inside Macintosh:Processes](#)

Translation Manager

The Translation Manager provides file and scrap translation to allow data to be used by applications other than those that originally created the data. The following changes have been made in the Translation Manager:

- The Translation Manager implementation has been moved to the System file, and the user interface has been moved from the now-obsolete Mac OS Easy Open control panel to the File Exchange control panel's "File Translation" panel.
- The Translation manager is now implemented in PowerPC code.
- The available translator lists are now rebuilt dynamically. If a translation extension component is registered sometime after startup, the Translation Manager will recognize it immediately.
- Contents of the "Translated Documents" folder are now removed at system startup time. The Translation Manager places automatically translated documents into this folder before passing them on to applications.
- With both QuickTime and MacLinkPlus for Easy Open installed, there are more than 64 possible translation configurations for Desktop Translators, but Desktop Translators could only be configured to use one of the first 64 translations available. This limitation for Desktop Translators has been removed.
- Applications which open files of a given type may do so without translation. Prior to File Exchange 3.0, if an application claimed (through its bundle and/or 'open' resource) that it could open files of a given type, documents of that type with a different creator type than the application's creator type would be translated rather than opened directly. Now, such documents are opened directly without translation. To see an example of this behavior, under Mac OS 8.1 drop a MoviePlayer movie onto SimpleText. It will open as a picture rather than a movie (as it will be translated by the QuickTime translator first). Now, the movie will open as expected.
- Translation Manager no longer creates a temporary heap zone for translation extension components. Instead, translation extension components are loaded into the system heap. This means that translation extensions are no longer allowed to "leak" memory, as their calls to `NewPtr` and `NewHandle` now operate in the system heap. Contrary to what is stated in [Inside Macintosh: More Macintosh Toolbox](#), the Translation Manager now disposes of the translation advertisement when tearing down the translation progress dialog (since the translation advertisement is now allocated in the system heap rather than a temporary heap).

COMPATIBILITY NOTE

Developers of PowerPC translation extensions are advised to package them as specified in Technote [TN1004](#), rather than as is done in the Mac OS Easy Open SDK.

Related Materials:

- The [File Exchange](#) control panel section
- Technote [TB41](#), "Translation Manager 1.1"
- The [Translation Manager](#) chapter of *[Inside Macintosh: More Macintosh Toolbox](#)*
- The [Translation Manager SDK](#)

Virtual Memory Manager

The Virtual Memory Manager provides virtual memory services for the Mac OS. The following changes have been made in the Virtual Memory Manager:

- All code needed to run VM on 68040 processors has been removed.
- `GetPageState` now correctly returns the state of pages in file-mapped space. Before Mac OS 8.5, `GetPageState` returned `kPageOnDisk` for unmapped pages. With Mac OS 8.5, `GetPageState` returns `kNotPaged` for unmapped pages (as it does for all other memory not controlled by the Virtual Memory).
- `DeferUserFn` previously returned a garbage result if the user function was called immediately. It now returns `noErr`.

Related Materials:

- The [DriverServicesLib](#) and [Memory Manager](#) sections
- Technote [TN1094](#), "Virtual Memory Application Compatibility"
- Technote [TN1104](#), "Interrupt Safe Routines"
- Technote [IM_ERRATA 02](#), "Inside Macintosh -- Memory Errata"
- Q&A [DV32](#), "PrepareMemoryForIO and Execution Levels"
- The [Memory](#) Technote Collection
- The [Memory](#) Q&A Collection
- The [Virtual Memory Manager](#) chapter of *[Inside Macintosh: Memory](#)*
- Minow, M. [The New Device Drivers: Memory Matters](#). [Develop](#) 24

Window Manager

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

- The Window Manager is now implemented in PowerPC code.
- An extensible, collection-based window resource format ('wind') has been added.
- Floating window support has been added with the routines `CreateNewWindow`, `GetWindowClass`, `ShowFloatingWindows`, `HideFloatingWindows`, and `AreFloatingWindowsVisible`.
- Extensive error and parameter checking has been added to all routines.
- New, advanced window positioning support routines have been added, including `SetWindowBounds`, `GetWindowBounds`, `MoveWindowStructure`, `ZoomWindowIdeal`, and `IsWindowInStandardState`.
- Support for associating tagged data with windows (properties) has been added.
- `GetWindowRegion` is a new call that always returns the visible state of the specified region, even for invisible windows.
- Update events will no longer occur in foreground application windows when overlapping windows in background applications become visible.
- A window's background can now be set to a specific color or pattern without the use of a 'wctb' resource.
- Routines for modifying a window's update region without changing the current port have been added (`InvalidWindowRgn/Rect`, `ValidWindowRgn/Rect`).
- Support for window title popup menus has been added.
- The `TransitionWindow` routine has been added for attaching animation and sound to a window for different on-screen actions such as showing or hiding a window.
- Under some circumstances, parts of the desktop were not being redrawn correctly. This has been corrected.
- Support for window proxies has been added.
- Window titles are now drawn using condensed and truncated text if the window's title is too large to be displayed in the title bar.

WARNING:

Applications using the undocumented and unsupported Layer Manager interface for floating windows will not work with Mac OS 8.5.

WARNING:

Applications writing directly to the low-memory window list global, either by direct access or by `LMSetWindowList`, may have problems in Mac OS 8.5. It is strongly recommended that developers interested in future compatibility treat the window list as read-only.

WARNING:

Applications writing directly to the `GrayRgn` may cause problems in Mac OS 8.5, and may cease to work properly in future versions of Mac OS. Developers interested in future compatibility should use the new `ShowMenuBar/HideMenuBar` routines instead.

Related Materials:

- The [Appearance Manager](#), [Control Manager](#), [Dialog Manager](#), and [Menu Manager](#) sections
 - Technote [TB39](#), "Appearance Versions"
 - The [High Level Toolbox](#) Technote Collection
 - The [High Level Toolbox](#) Q&A Collection
 - The [Window Manager Reference](#) chapter of *[Inside Macintosh: Mac OS 8 Toolbox Reference](#)*
 - The [Window Manager](#) chapter of *[Inside Macintosh: Macintosh Toolbox Essentials](#)*
 - *[Inside Macintosh: Macintosh OS 8 Human Interface Guidelines](#)*
 - The [Appearance 1.0.3 SDK](#)
-



Control Panels

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

In Mac OS 8.5, several related control panels have been consolidated into individual control panels.



Appearance 1.1

The Appearance control panel provides a user interface for configuring the appearance of graphical user interface elements.

- The new Appearance control panel provides several new panels for creating and modifying themes.
- Options previously available through the Desktop Pictures control panel are now provided by the Appearance control panel.
- AppleScript support includes:
 - Query and set every Appearance option (theme, color, sound effects, desktop pictures and patterns, fonts, etc.)

Related Materials:

- The [Appearance Manager](#), [Control Manager](#), [Dialog Manager](#), [Menu Manager](#), and [Window Manager](#) sections
- Q&A [TB39](#), "Appearance Versions"
- The [Appearance Manager Reference](#) chapter of *Inside Macintosh: Mac OS 8 Toolbox Reference*
- *Inside Macintosh: Macintosh OS 8 Human Interface Guidelines*
- The [Appearance 1.0.3 SDK](#)



Apple Menu Options 1.1.4

The Apple Menu Options control panel provides user interface configuration facilities for the appearance and behavior of the Apple Menu. There are some corrections present in this implementation of the Apple Menu Options control panel.

- A problem occurred where the Apple Menu was not being drawn correctly when the system font was changed to a font of a different size. This has been corrected.
- Balloons no longer appear when the Apple Menu Options control panel is frontmost and the mouse is not over its window.
- A problem where Apple Menu Options' patch to `UnmountVol` could cause a big disk thrash has been corrected. This would happen if the volume passed to `UnmountVol` is not identified by a volume reference number.
- AppleScript support includes:
 - Query and set all Control Panel settings, including count of recently opened items to remember.



ColorSync 2.5

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

- Multiprocessing capabilities have been added.
- AppleScript support for the following has been added:
 - Embedding a ColorSync profile in an image file.
 - Match or proof an image to an output device.
 - Query or set the default profile.

Related Materials:

- [Inside Macintosh: Advanced Color Imaging on the Mac OS](#)
- Hayward, D. [PrintingHints: Syncing Up With Colorsync 2.0](#). [Develop](#) 23
- ColorSync web pages at <http://colorsync.apple.com/>
- The [ColorSync 2.5.1 SDK](#)



Control Strip 2.0

Control Strip is a control panel providing configuration facilities for the control strip.

- Control Strip 2.0 is now an application ('APPC') and contains fewer trap patches.
- Control Strip 2.0 corrects a number of problems with the previous version and adds several features that expand the user experience.
- Version 2.0 adds module drag-and-drop install capability. Control strip modules can now be installed by dragging and dropping the module into the control strip.

Related Materials:

- Technote [OS06](#), "Control Strip Modules"
- Q&A [OPS15](#), "Calling Control Strip Routines from PowerPC Code"
- The [Control Strip SDK](#)



Date & Time 8.1

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.

- Internet time synchronization by way of Network Time Protocol (NTP) and automatic daylight savings time support have been added.
- Location information is now set in the Date & Time control panel.
- Added Regina, Saskatchewan, Canada to the list of time zone locations.
- Fixed incorrect spelling of the city name "Tientsin" to "Tianjin".

Related Materials:

- Technote [TN1049](#), "Approaching the Millennium: The Mac and the Year 2000"
- Q&A [OPS01](#), "System Clock"
- Technote [TN1064](#), "Inside Macintosh: Operating System Utilities: Addendum to Chapter 4 -- Determining If Daylight Savings Time is in Effect"
- The [Date, Time, and Measurement Utilities](#) chapter of *[Inside Macintosh: Operating System Utilities](#)*
- Minow, M. [Timing on the Macintosh](#). [Develop](#) 29



File Exchange 3.0

File Exchange allows users to set mappings from filename extensions (ex.: ".TXT") to Mac OS file types (ex.: "TEXT"), to mount DOS SCSI disks and drive containers, and to set File Translation preferences. The following changes have been made in File Exchange:

- File Exchange Features
 - PowerPC-native.
 - Consolidated interface for PC Exchange and File Translation (formerly Mac OS Easy Open).
 - Supports AppleScript.
 - Integrated editing of Internet Config extension mappings.
- PC Foreign File System Features
 - PowerPC-native.
 - Mounts all available PC volumes at startup.
 - Can also mount all available PC volumes "on the fly".
 - Supports multiple SCSI busses.
 - Uses HDI driver (from DiskCopy) to mount PC disk images.
 - Improved read/write/cache code path.

- Internet Config integration.
- Desktop Database support.
- Permits File Sharing of PC volumes and disk images.
- AppleScript support includes:
 - Query and create translation and extension mappings.
 - Query or change control panel settings.
- Corrected a problem where a particular data file format saved to PC disk was unreadable. The error returned by `SetFPos` was not being handled correctly when an attempt to set the mark past EOF was being made.
- Two-byte characters in traditional DOS FAT "8.3" format file names weren't being displayed in Get Info. This has been corrected.
- Determines disk type (FAT12, FAT16, FAT32) using disk geometry; this fixes problems with low-capacity FAT16-formatted PC cards.
- The PC foreign file system's "mimicry" of HFS has been improved. Now, under as many circumstances as possible, the PC foreign file system behaves the same as HFS.
- Allows use of PC disks with allocation block size of 32K or larger (for example, 2 GB disks).

Related Materials:

- The [Translation Manager](#) section



File Sharing 8.0

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

- AppleScript support includes:
 - Query or change control panel settings.
 - Disconnect a user.

Related Materials:

- The [File System Manager](#), [File Manager](#), and [AppleShare Workstation Client](#) sections
- The [Networking](#) Technote Collection
- The [File Sharing](#) section in Technote [TN1102](#), "Mac OS 8"
- Technote [TN1111](#), "Programmatic Mounting of AppleShare Volumes"
- Technote [FL20](#), "File Sharing and Shared Folders"
- Technote [FL28](#), "PBShare, PBUnshare, and PBGetUGEntry"
- Technote [FL37](#), "You Want Permission to do What?!!"
- AppleShare IP web pages at <http://www.apple.com/appleshareip/>



Internet 1.0.1

A new control panel application ('APPC') which provides user interface for creating and editing [Internet Config](#) preferences, including support for switchable preference sets for handling multiple locations or users. Provides AppleScript support utilizing some of this configuration information, including the following commands:

- Go to a URL
- Read Mail
- Related Materials:
 - The [Internet Config](#) section
 - The Internet Config web pages at <http://www.quinn.echidna.id.au/Quinn/Config/>



Keyboard 8.1

The Keyboard control panel provides a user interface for configuring connected keyboards.

- A control for turning on or off font and keyboard synchronization has been added.
- The Keyboard menu now correctly displays balloon help for items beyond the seventh.
- Balloon help messages for the keyboard menu items have been reworded in some cases.
- The keyboard menu now has a "Customize..." menu item as the last menu item. Choosing it will open the Keyboard control panel.

Related Materials:

- The [QuickDraw Text](#) and [Script Manager](#) sections



Location Manager 2.0

Apple Location Manager is an extension that allows mobile users to save and restore sets of configurations ("locations") under a single name. For example, a user can define a group of printers, network settings, and extension sets for different computing environments.

- The maximum number of locations has been increased.
- A problem where the system startup screen was not being redrawn correctly after the Location Manager extension window was dismissed has been corrected.
- AppleScript support includes:
 - Query or change the current location.

Related Materials:

- The [Location Manager web pages](#)
- The [Location Manager 1.0.1 SDK](#)



Memory 8.0

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 Memory control panel is now an 'APPC'.
- Fully supports the Appearance Manager.
- By default, the disk cache size is now automatically calculated. Users can still customize the disk cache size setting, or they can allow the system to calculate a default setting for them.

Related Materials:

- The [Memory Manager](#) and [Virtual Memory Manager](#) sections



Monitors & Sound 1.4.2

The Monitors & Sound control panel provides a user interface for configuring and adjusting various settings for AppleVision monitors attached to the computer, along with services formerly provided by the Monitors control panel and by the Sound control panel.

- The Monitors and Sound control panel has been updated to support new hardware.



Remote Access 3.1

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

- ARA 3.1 includes support for both TCP/IP and AppleTalk (ATCP) over a PPP link.
- Apple Remote Access 2.1 and OT/PPP 1.0.1 have been replaced by ARA 3.1; version 3.1 will now be the default, consolidating remote client software supporting both PPP and ARAP protocols.
- The Remote Access and PPP control panels have been consolidated into the Remote Access control panel.
- ARA 3.1 contains a few feature enhancements and many bug fixes since ARA 3.0, ARA 2.1, and OT/PPP 1.0.1.
- The PPP client now supports MS-CHAP and includes other improvements for better Windows NT compatibility.
- This release includes the latest modem scripts for Apple and popular third-party modems.

Related Materials:

- The [OT/PPP SDK](#)



TCP/IP 2.0.1

The TCP/IP control panel provides a user interface for configuring TCP/IP protocols.

- To cut down the amount of typing required to manually configure TCP/IP, the "IP Address" field now allows addresses to be entered in "address/mask" notation. For example, if the address "12.34.56.78/24" is entered, the 24-contiguous-ones mask value (255.255.255.0) is automatically entered in the "Subnet mask" field, and the "Router address" field, if empty, is automatically initialized to the first valid address on that subnet (12.34.56.1 in this example). The "Subnet mask" and "Router address" fields remain editable should the user wish to change the automatically entered values.
- A new field has been added allowing users to enter their DHCP "client id".

Related Materials:

- The [OpenTransport](#) section



Users and Groups 8.0.1

This control panel allows the user to designate which people are allowed to establish network connections with the computer.

AppleScript support includes:

- Query or set privileges of users and groups.
- Create and delete users and groups.



Web Sharing 1.5.1

Web Sharing allows users to configure and set up the "Web Sharing" folder on their computer to act as a World Wide Web site. Web Sharing runs as an HTTP server, allowing users to access files stored in the Web Sharing folder using any web browser program available on any computer platform. New features for Web Sharing include:

- Added human interfaces for configuring MIME types.
- Connection logging facilities have been added.

Related Materials:

- The Personal Web Sharing web pages at:
<http://www.claris.com/products/apple/personalsharing/personalsharing.html>



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.

The "Extensions Disabled" message has been replaced with the text "Extensions Off" for consistency with documentation and terminology.



Apple CD/DVD Driver 1.0.1

The Apple CD-ROM driver contains some enhancements for support of new Apple products and corrects some problems found in previous versions of the driver.

- Apple CD-ROM and Apple DVD-ROM drivers have been combined into a single file called Apple CD/DVD Driver.

COMPATIBILITY NOTE

Non-Apple CD drives are not supported by the Apple CD-ROM driver.

Related Materials:

- Technote [DV18](#), "CD-ROM Notes (Most Excellent)"
- Technote [DV22](#), "CD-ROM Driver Calls"
- Technote [DV505](#), "CD-ROM QAs"
- Technote [FL36](#), "Apple Extensions to ISO 9660"
- Q&A [DV18](#), "Detecting a CD-ROM"
- Q&A [OPS05](#), "Disconnected aliases on CD-ROM and Alias Manager"



Apple Enet 2.0

The Apple Ethernet driver provides support for Apple's built-in Ethernet and cards on newer machines. Features include:

- Improved performance.
- Obsoletes a number of older drivers and consolidates their functionality into one file.

Related Materials:

- The [OpenTransport](#) section
- A new Ethernet Driver framework is available that replaces the Mentat template
- The [Open Transport SDK](#)



AppleScript 1.3.2

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.

- AppleScript is now PowerPC-native. Raw AppleScript execution is up to five times faster as compared to the same script run under Mac OS 8.1.

- **Unit Types and Coercions (built-in)**

This version introduces the concept of units of measurement and some basic units. Applications or scripting additions can extend these easily. The types defined include:

- length
- area
- volume (liquid and solid)
- temperature

A complete list of specific types can be found in the Help Center in the AppleScript section. (Note: the Help Center claims AppleScript has support for quarts; this is not the case.) Constants can be found in `<ASRegistry.h>`.

The basic manner of use in AppleScript scripts is by coercion, e.g. "5 as inches". A value of one class can be coerced into another similar class, such as "5 as feet as inches", so long as both classes share a common unit type (in this case, length, whose base unit is meters).

To compile the displayed results of a command that returns a unit-type value, a global `Get Data` handler has been added to resolve the unit type object specifier and coerce it to a value.

- **Unicode Types and Coercions (built-in)**

This version introduces the type `Unicode text`. Its basic structure is a stream of Unicode text. It can be coerced to and from the normal `text`, `styledText`, and `int1Text`.

- **Consolidated Scripting Additions**

The following scripting additions, which previously were delivered as separate files in previous versions of Mac OS, have been consolidated into the "Standard Additions" scripting addition:

- AGStart
- Beep
- Choose Application
- Choose File
- Current Date
- Display Dialog (now allows you to specify a timeout)
- File Commands
- Load Script
- Numerics
- String Commands
- Read/Write Commands
- Run Script
- Store Script
- Time To GMT

COMPATIBILITY NOTE

The interface and functionality of the consolidated scripting additions listed above is unchanged from AppleScript 1.1.2.

● New Suites

- New suites for Internet data types, Mac OS connectivity, and folder actions are now included in the dictionary for the Standard Additions scripting addition.

● New Scripting Additions

Several new scripting addition commands are added. The following lists the new additions. For more information, see the AppleScript 1.3 release notes.

- Choose: an item from a list in a dialog
- Delay: for a fixed amount of time
- Say: the given text
- Summarize: the specified string or file in a short block of text. Uses Sherlock technology.
- Mount volume: an AppleShare server volume
- The Clipboard scripting addition from Jon's Commands was integrated into the Standard Additions.

● Corrections and Bug Fixes

- The Read/Write Commands can now read lists of lists correctly.
- Displaying certain large, nested, or recursive lists could cause a crash. This has been fixed by adding processor stack checks.
- The Store Script scripting addition was leaving its resource fork open after it had completed execution. Under some conditions, subsequent calls to the Resource Manager could load resources from AppleScript into the current application's heap, causing unexpected problems. This has been corrected, and the Store Script scripting addition closes its resource file after use.
- A memory leak in Run Script was fixed.
- AppleScript Gestalt and Component versions are now correct; interface is 1.1 and implementation is 1.3.
- The AppleScript "Applet" component was clearing one byte of memory at location \$0006. This has not been a problem on Mac OS computers, as this location is normally already 0.

Related Materials:

- The [Finder: AppleScript Support](#) section
- The discussion of AppleScript in chapters 7 ([Introduction to Scripting](#)), 8 ([Apple Event Terminology Resources](#)), 9 ([Recording Apple Events](#)), and 10 ([Scripting Components](#)) of *Inside Macintosh: Interraplication Communication*
- [Inside Macintosh: AppleScript Finder Guide](#)
- [Inside Macintosh: AppleScript Language Guide](#)

- o [Inside Macintosh: AppleScript Scripting Additions Guide](#)
- o The AppleScript web pages at <http://AppleScript.apple.com/>
- o AppleScript 1.3 Release notes
- o Simone, C. [According to Script: User Interactions in Apple Event-Driven Applications. Develop](#) 29
- o Simone, C. [According to Script: Attaching and Embedding Scripts. Develop](#) 26
- o Simone, C. [According to Script: Properties and Preferences. Develop](#) 25
- o Simone, C. [According to Script: Steps To Scriptability. Develop](#) 24
- o Anderson, G. [Speeding Up whose Clause Resolution in Your Scriptable Application. Develop](#) 24
- o Simone, C. [According to Script: Thinking About Dictionaries. Develop](#) 23
- o Anderson, G. [Scripting the Finder. Develop](#), 20
- o The [AppleScript SDK](#)



AppleShare Workstation Client 3.8

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

COMPATIBILITY NOTE

AppleShare Workstation Client 3.8 can be used with Mac OS 7.5.3 and later.

- A new UAM (User Authentication Module) API has been added that allows UAMs to use TCP/IP.

Related Materials:

- o The [File Sharing](#) section
- o The [Networking](#) Technote Collection
- o The [File Sharing](#) section in Technote [TN1102](#), "Mac OS 8"
- o Technote [TN1111](#), "Programmatic Mounting of AppleShare Volumes"
- o Technote [FL20](#), "File Sharing and Shared Folders"
- o Technote [FL28](#), "PBShare, PBUnshare, and PBGetUGEntry"
- o Technote [FL37](#), "You Want Permission to do What?!!"
- o AppleShare IP web pages at <http://www.apple.com/appleshareip/>



Application Switcher 1.0

The Application Switcher is a new user interface element that appears as a floating window, providing a clickable list of active processes running on the computer.

COMPATIBILITY NOTE

Application Switcher uses command-tab as a keyboard shortcut for switching between open applications. It is possible to disable or change the keyboard shortcut for switching between open applications using the AppleScript script on the Application Switcher Help page.

AppleScript support for the application window includes:

- Query and set location, size, and properties of the Application Window



Desktop Printing 2.2.1

Desktop Printing provides easy access to printers by locating them on the desktop. Users can access information about their print jobs by opening these icons, and they can print documents by dropping them into the printer icon.

- Third party desktop printing is now supported.

AppleScript support includes:

- Get a list of all desktop printers.
- Create a new desktop printer.
- Set the current printer to an existing desktop printer.

Related Materials:

- The [LaserWriter 8](#) section
- Technote [TN1097](#), "Desktop Printing Revealed"
- Technote [TN1113](#), "Customizing Desktop Printer Utility"
- Technote [TN1131](#), "Creating Desktop Printers on the Fly"



Find By Content 2.0

Provides system-wide search facilities for finding related files based on their content.

- A CFM-based interface is available for developers wanting to access the Find By Content services to search files on disks from within their applications.

Related Materials:

- Technote [1141](#), "Extending and Controlling Sherlock and the Find by Content Libraries"
- The [Find By Content SDK](#)



Internet Config 2.0.2

Internet Config provides a centralized storehouse of information for various settings utilized by Internet applications. It provides methods for applications to both retrieve and store configuration information.

- Support for switchable sets of preferences has been added for handling multiple locations or users.
- A new API for interrupt-safe file extension mapping designed for file system developers has been added.
- Additional file types have been added to the file extension mapping database.
- Internet Config clients can now link against a CFM library.

Related Materials:

- The [Internet](#) control panel section
- Technote [TN1004](#), "On QuickTime Component Manager 3.0 & PowerPC Native Components"
- Technote [QT05](#), "Component Manager version 3.0"
- The [Component Manager](#) chapter of *Inside Macintosh: More Macintosh ToolBox*
- The Internet Config web pages at <http://www.quinn.echidna.id.au/Quinn/Config/>



LaserWriter 8 version 8.6

LaserWriter 8 provides the ability to print to PostScript printers. The new LaserWriter 8 software provides many enhancements and new features of interest to developers including the following:

- LaserWriter driver supports Unicode and the Euro character.
- Support for printing Unicode text.
- Better support for Desktop Printing.
- Support for the Appearance Manager.
- Support for ColorSync 2.5.
- Improved facilities for printing of PS, EPS, JIFF/JPEG, and PICT file formats.
- Support for printing over a variety of communication links including IrDA, PAP, LPR, and developer-defined communication channels ("custom hoses").
- LaserWriter 8 now uses the 'ppdf' FindFolder selector to locate printer description files.
- Desktop Printer files can now be conveniently shared among users on different computers.
- Support for partial TrueType font downloading has been added. This feature is dependent on ATSUI and supersedes the TrueType Booster Extension found in some systems using 2-byte characters.

Related Materials:

- o The [Desktop Printing](#) section
- o The [Printing](#) Technote Collection
- o The [Printing](#) Q&A Collection
- o Technote [1140](#): "The Euro Currency Symbol"
- o Technote: "Introducing the LaserWriter 8 Driver Version 8.6" (coming soon)
- o The [Printing Manager](#) chapter of [Inside Macintosh: Imaging with QuickDraw](#).
- o Hayward, D. [Print Hints: Syncing Up With ColorSync 2.0](#). [Develop](#) 23
- o Gelphman, D. [Printing Images Faster with Data Compression](#). [Develop](#) 24
- o Polaschek, D. [Print Hints: The Top 10 Printing Crimes Revisited](#). [Develop](#) 26
- o Polaschek, D. [Print Hints: The All-New LaserWriter Driver Version 8.4](#). [Develop](#) 27
- o Polaschek, D. [Print Hints: Safe Travel Through the Printing Jungle](#). [Develop](#) 28
- o Polaschek, D. [Print Hints: Sending PostScript Files to a LaserWriter](#). [Develop](#) 29
- o Apple's Printing developer pages at [</macos/printing.html>](http://macos/printing.html)



Mac OS Runtime for Java 2.0

Mac OS Runtime for Java (MRJ) is Apple's implementation of the Java runtime environment. MRJ allows Java functionality to be embedded in Mac OS applications, and applications using MRJ's JManager API are able to run Java applets and applications.

- Related Materials:
 - o [Inside Macintosh: Programming With MRJ Toolkit](#)
 - o [Inside Macintosh: Programming With JManager For 1.0.1](#)
 - o [Inside Macintosh: Programming With JManager For 2.0](#)
 - o [Inside Macintosh: Using JBindery 2.0.1](#)
 - o Main website: [<http://www.apple.com/macos/java/>](http://www.apple.com/macos/java/)
 - o [Java Developer website](#)
 - o The [Mac OS Runtime for Java SDK](#)



Network Setup Extension 1.0

The Network Setup Extension provides for programmatic configuration of the AppleTalk, TCP/IP, Infrared, and Remote Access settings.

- The Network Setup Extension provides both AppleScript and CFM-based APIs for configuration of network settings.



Open Transport 2.0.1

Open Transport is the networking technology for Mac OS 8.5. Open Transport 2.0.1 contains SNMP and OT SNMP Admin, DHCP enhancements for improved compatibility with Windows NT, including Client ID, new APIs for manipulating configurations, scriptability for AppleTalk, TCP, Modem, Remote Access, Infrared control panels, and bug fixes.

- The Dynamic Host Configuration Protocol (DHCP) implementation now supports a number of much-requested features, including Client Identification, retention of the lease on a given IP address across boots, better user feedback in the event that the DHCP server is unavailable, and better support of RFC 2131. Many of these changes address customer concerns about OT's DHCP compatibility with Windows NT DHCP servers.
- Support for the Simple Network Management Protocol (SNMP) is available for Open Transport. Support for the version 2 Management Information Base (MIB II), instrumentation for AppleTalk and TCP/IP protocols and transports, administrative tools, and system agent support for the PCI bus will be provided.
- Minor changes to Open Transport support for Apple Remote Access to allow feature requests such as support for MS-CHAP Authentication support and RFC 1877 to automatically negotiate DNS address with a server. These changes address customer concerns about OT's compatibility with Windows NT PPP servers.
- If Open Transport TCP/IP is configured to use DHCP and it receives no response from any DHCP server within 30 seconds, it will dynamically assign a temporary address in the local non-routable self-assignment range (169.254.x.x) and operate using this address until a DHCP server becomes available. This allows hosts on the same Ethernet to communicate with each other using IP (for example, to allow high-performance TCP/IP access to an AppleShare IP server) even if no connection to the Internet is available.

Related Materials:

- [*Inside Macintosh: Networking With OpenTransport*](#)
- The [OpenTransport web pages](#)
- The [OpenTransport SDK](#)
- The [OpenTransport/PPP SDK](#)



PlainTalk 1.5.3

PlainTalk provides both speech recognition services and services for the translation of text into audible speech. Mac OS 8.5 installation ships with English Text to Speech 1.5.3, Speech Recognition, and Mexican Spanish Text to Speech. English Text to Speech is installed as part of the default Mac OS 8.5 installation. Speech Recognition can be installed as a custom install option.

- Speech is now available via the Say scripting addition -- part of the standard AppleScript additions.
- The new "Read Me" file contains information on how to place emphasis on certain words that are spoken.
- Three new high-quality voices have been added that were previously available as a separate download from the speech web pages.
- Talking alerts are now turned on in the default install (with a 10-second delay).

Related Materials:

- The [Speech Manager](#) chapter of *Inside Macintosh: Sound*
- Pallakoff, M. & Reeves, A. [The Speech Recognition Manager Revealed](#). [Develop](#) 27
- Monroe, T. [Adding Speech Recognition to an Application Framework](#). [Develop](#) 27
- The Speech web pages at <http://www.apple.com/speech>
- The [Speech Recognition Manager 1.5.1 SDK](#)
- The [Speech Synthesis Manager 1.5 SDK](#)



QuickTime 3.0

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.

- Related Materials:
 - [Inside Macintosh: QuickTime](#)
 - [Inside Macintosh: QuickTime Components](#)
 - The [QuickTime web pages](#)
 - The [QuickTime 3 Developer Documentation web page](#)



QuickDraw 3D 1.5.4

QuickDraw 3D provides 3D drawing services for the Mac OS.

Related Materials:

- [*Inside Macintosh: 3D Graphics Programming with QuickDraw 3D*](#)
 - [*Inside Macintosh: 3D Metafile Reference*](#)
 - Thompson, N. [Easy 3D With the QuickDraw 3D Viewer](#). [Develop](#) 29
 - Schneider, P. [New QuickDraw 3D Geometries](#). [Develop](#) 28
 - McBride, P. [Game Controls for QuickDraw 3D](#). [Develop](#) 27
 - Falco, P. & McBride, P. [Generating QuickTime VR Movies from QuickDraw 3D](#). [Develop](#) 25
 - Thompson, N. & Fernicola, P. [Graphical Truffles: Making the Most of QuickDraw 3D](#). [Develop](#) 24
 - Thompson, N. & Fernicola, P. [The Basics of QuickDraw 3D Geometries](#). [Develop](#) 23
 - Fernicola, P. & Thompson, N. [QuickDraw 3D: A New Dimension for Macintosh Graphics](#). [Develop](#) 22
 - The QuickDraw 3D web pages at <http://www.apple.com/quicktime/qd3d/index.html>
-



Apple Menu Items

Items in the Apple Menu Items folder appear in the Apple Menu. Files of type 'APPD' are auto-routed to the Apple Menu Items folder when they are dropped on the System Folder's icon.



Apple System Profiler 2.1.1

The Apple System Profiler 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.

- AppleScript support includes:
 - Query or set which data options to gather in a system profile report.
 - Create a system profile report (document).
 - Save a report to disk.
 - Print a report.
- Added a close box to Quit the application.
- The Apple System Profiler now searches and reports more hardware and software configuration information than did previous versions.



Key Caps 8.1

Key Caps provides a user interface for displaying and viewing the characters produced by various key combinations.

- Key Caps will work with System 7.5.3 and later versions of the Mac OS.
- Key Caps is now an 'APPD'-type application.
- Key Caps provides Unicode support for 'uchr' system resources.
- For better display of multiple Unicode characters that map to a single key, as in the Indic and Unicode keyboard layouts, the Key Caps window can now be zoomed to a larger size. When the cursor is on a key that displays multiple Unicode characters, a yellow floating popup window showing all the characters will appear below the key.
- Dead keys are outlined using a gray color instead of a gray pattern.
- Support for inline text entry has been added.
- KeyCaps supports 'fn' and 'num-lock' mode display on PowerBook G3 computers.
- To allow for better character visibility, Key Caps now draws small keys with a smaller font and less 3D styling than other square keys. For example, the arrow keys on the PowerBook G3 are drawn in this way.
- Any new keys with either the standard 21-pixel width or 21-pixel height (defined in 'KCAP' resources) will be correctly drawn without requiring any new supporting 'kbit' or 'kptr' resource in Key Caps.
- The Key Caps Preferences file in the Preferences folder now contains the version of the last instance of Key Caps that wrote into the Preferences file.
- A problem where a crash could occur when pasting a large amount of text into the text box has been corrected.

Related Materials:

- Technote [TE27](#), "Inline Input for TextEdit with TSMTE"
- Technote [TE515](#), "Keyboard Resource QAs"
- The [Storing a Font Name in a Document](#) section of the [Font Manager](#) chapter of [Inside Macintosh: Text](#)
- The [Key Caps Desk Accessory](#) section of [Appendix C - Keyboard Resources](#) in [Inside Macintosh: Text](#)



Network Browser 1.0

- The Network Browser application allows for convenient access to file servers available on AppleTalk networks.



Sherlock 2.0.1

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.

- Sherlock passes additional information to applications along with 'odoc' events, allowing applications to display information about how a file was found.
- Sherlock utilizes search plug-in files describing Internet search sites. Specifications are available for search site administrators to create their own search plug-in files, thereby making their search sites available to the Sherlock application.
- Sherlock provides AppleScript support for searching by content, indexing volumes, searching the Internet, and searching for files by name.

Related Materials:

- Technote [1141](#), "Extending and Controlling Sherlock and the Find by Content Libraries"
 - The [Find By Content SDK](#)
-

Apple Extras



Apple Video Player 1.7.1

The Apple Video Player provides facilities for live video playback.

- AppleScript support includes:
 - Play video from any source.
 - Capture video to QuickTime movie or PICT.
 - Capture Teletext or Closed Caption data.
 - Navigate Teletext screens.
 - Change channel, volume, Teletext page, bass/treble, brightness/sharpness, etc.



SNMP Administrator 1.0

Open Transport now supports SNMP (Simple Network Management Protocol). SNMP Administrator provides a user interface for configuring SNMP.

- AppleScript support includes the ability to address and send e-mail.

Related Materials:

- Technote [NW19](#), "How the Simple Network Management Protocol (SNMP) Manager Finds Network Cards"
 - Technote [NW22](#), "SNMP Transports"
 - The [MacSNMP 1.1.1 SDK](#)
-

Utilities



Disk Copy 6.3.1

Disk Copy allows for disk image creation, mounting, and writing.

- AppleScript support includes:
 - Query and set all application preferences.
 - Mount a Disk Copy image as a disk volume.
 - Unmount Disk Copy volumes.
 - Check and verify Disk Copy images and checksums.
 - Make a floppy from a Disk Copy document.



Disk First Aid 8.5.1

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.

- Code from Disk First Aid runs whenever the "improper shut down" dialog appears.
- Unlike previous versions of Disk First Aid, the new Disk First Aid program allows you to repair the startup volume and other volumes with open files.
- Disk First Aid now verifies and repairs the "HFS wrapper" on HFS Plus volumes.
- Disk First Aid can now repair many kinds of damage caused by older disk utilities that are not compatible with HFS Plus volumes.

AppleScript support includes:

- Verify and/or repair a volume.

Related Materials:

- The Disk First Aid "Read Me" file



Drive Setup 1.6

Drive Setup provides drive initialization services for Apple hard drives.

- Drive Setup can now reinitialize a drive maintaining the former partition scheme and formats.
- Drive Setup warns before updating a driver.
- Contains bug fixes.

Apple Developer Utilities

Mac OS 8.5 requires updated copies of the following Apple Developer Utilities:



MacsBug 6.5.4a6

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

COMPATIBILITY NOTE

Mac OS 8.5 will not load versions of MacsBug earlier than 6.5.4a5. If Mac OS 8.5 finds an unsupported version of MacsBug in the System Folder, it will attempt to rename the MacsBug file to "MacsBug Obsolete" (where " Obsolete" is a localized string).

COMPATIBILITY NOTE

In Mac OS 8.5, the Font Manager has its own private heap zone. This heap has no free blocks. As a result, Macsbug 6.5.4 reports the heap is corrupt. This is a harmless condition, not a real corrupt heap; Macsbug is reporting the wrong thing.

Related Materials:

- MacsBug Reference and Debugging Guide
- Download [MacsBug](#)

Macintosh Programmer's Workshop (MPW)

Some versions of MPW Shell and StdCLib can cause problems under Mac OS 8.5. Click [here](#) for more information.

Items No Longer Supported

Appearance Extension

The Appearance Extension is no longer in the Extensions folder. Its functionality has been incorporated into the System file.

Apple Remote Access Client

Apple Remote Access Client has been integrated into Remote Access 3.1.

Cyberdog

Cyberdog is no longer included with Mac OS. For more information, see the OpenDoc web pages at <http://www.opendoc.apple.com/>.

Desktop Pictures control panel

The Desktop Pictures control panel is no longer installed. Its functionality has been incorporated into the Appearance control panel.

Mac OS Easy Open

The Mac OS Easy Open control panel is no longer installed. Its interface has been incorporated into the File Exchange control panel, and its functionality has been moved to the System file.

Map control panel

The Map control panel is no longer supported. Its functionality has been incorporated into the Date & Time control panel. For backwards compatibility, it is provided in the Apple Extras folder.

OpenDoc

OpenDoc is no longer included with the Mac OS. For more information, see the OpenDoc web pages at <http://www.opendoc.apple.com/>.

AppleShare IP 6.1 will require and include OpenDoc when it ships, and will be supported on Mac OS 8.5 in that configuration.

Open Transport/PPP 1.0.1

Open Transport/PPP has been integrated into Remote Access 3.1.

WorldScript Power Adapter

The World Script Power Adapter has been obsoleted by changes to Script Manager, the Font Manager, et al.

Downloadables



[Acrobat version of this Note \(230K\).](#)

To contact us, please use the [Contact Us](#) page.
Updated: 26-Oct-98

[Technotes](#)
[Previous Technote](#) | [Contents](#)