



Mac OS 8 (D11E4) Release Notes

These release notes describe some of the changes that have occurred since the previous developer release of Mac OS 8. There may be functionality and changes not described here. This document is provided to you as an aid in determining the functionality of this release and which bugs are already known by Apple. The seven-digit numbers (e.g. 1246744) are used internally for bug tracking.

• Appearance Manager (Themes)

Bugs Fixed:

- SetTheme now restores port on exit (1246744)
- Pen PixPat no longer uses same pat as back PixPat (1331809)
- Control directionality now masked for scrollbars (1339213)
- Popup controls now draw correctly when in Right-to-left orientation (1335344)
- Control Invisibility no longer asserts, just exits draw function (1338987, 1337889)
- Window variant codes now work for dialogs and floating windows (1263898, 1341133, 1331822)

Known Bugs/Problems In This Release:

A state may occur with some applications where the port pen or background pixpat ends up nil.

This may cause a later crash in some cases.

Window titles of some windows get clipped off when redrawn on update or uncollapse.

Control focus is being redone and should finally work correctly.

• Apple Technology (International)

Kotoeri

Bugs fixed since d11e1:

- 1340125 Kotoeri does not do kana-kanji conversion
- 1346425 Kotoeri Keyboard icon is wrong
- 1349013 Annoying DebugStr in KotoeriIMInitRoutine
- 1346218 Kotoeri palette disappears when zoomed, frame remains
- 1346221 Bad redraw when scrolling through character palette.

Known bugs / unimplemented features:

- Daruma returns kanji characters grouped into morphemes; Kotoeri should combine these morphemes into coarser-grained groups based on part-of-speech,

but currently does not do so.

- Kotoeri should call Daruma multiple times to get a more comprehensive list of candidates, but currently only calls Daruma once (so you may not see the Kanji you're looking for).
- Not all of the two-byte Roman punctuation and other special characters are correctly transliterated (the table is incomplete).
- The following menu items have not been implemented:

also non-functional

- Kotoeri help \
- Open Dictionary > the corresponding buttons on the command palette are
- Kotoeri preferences /
- Retrieve fixed Yomi
- Transliterate ...

Japanese Locale

Bugs fixed In This Build:

- 1355895 D11E4c5: "Japanese" locale file requires 'lobj' resources for 'itlc' and
- 1355900 D11E4c5: an assert when boot in Japanese primary
- 1335203 Added Mac OS 8 collate table to the end of 'itl2' table
- 1344067 Correct wrong region ID

What's New In This Build:

Add 'spfn' locale objects.

Japanese Fonts

No change since d11e2

Language Analysis

No change since d11e2

Daruma Engine

What's New In This Build:

Once changed grammar file structure (1338777), but backed to old structure since 1338777 isn't approved yet.

Fixed a bug that the analysis cannot be done when the same string is specified as the last analysis call (1344522)

Update some part of speech code of grammar data to follow dictionary update (1340462)

Build files is changed to define new condition "FOR_PTR_BASED_AE" (1327405)

Implementation:

The number of implemented APIs is not changed since D11E2.

Daruma Dictionaries

Bugs fixed since d11e2:

1340462 Update dictionary data to improve Daruma's conversion accuracy.

Dictionary Manager

No change since d11e2.

FBit Enabler

No change since d11e2

Daruma Test App

Changes since d11e2:

Completely rewrote test application so that it can be build with MPW.

Now, this application doesn't use CodeWarrior's SIOUX library.

Features:

Can convert Kana to Kanji, or Kanji to Kana. The conversion direction can be selected by a popup menu.

The input method for Kana or Kanji doesn't work well yet, so this application has three short Japanese sentences in it. User can select those sentence and convert it with the selected method.

• Code Fragment Manager

What's new in this build:

No API changes.

Known problems:

Moving DLLs into and out of folders still has some problems. You may get CFM assertions about usage counts becoming negative. The application file/folder are not searched other than with the app load.

• Communications Toolbox

The Communications Toolbox (CommToolbox) is provided in d11e4 as six shared libraries: CommResources, CTBUtilities, Connections, FileTransfers, Terminals and CommToolbox.

CommResources, CTBUtilities, Connections, FileTransfers, Terminals: These are all creeper libraries. That is, every exported call just sends a debugstr and returns null (when appropriate). They also contain a list of routine descriptors used by the 68k dispatcher routine.

CommToolbox: This library exports a piece of data which is actually the 68k dispatcher routine code for the CommToolbox. It uses the routine descriptors exported by the manager libraries to dispatch 68k calls to the appropriate library.

Obviously, applications relying on the CommToolbox functionalities will not work. However, applications for which these functionalities are not vital, should not crash. There has been a typo in the name of the dispatcher library. It should be "CommToolboxTrap" and not "CommToolbox". There is no technical issue with this mistake, but it should be changed for the next build.

The CommToolbox will be submitted separately from the Open Transport team after DR1.

• Finder

Aliases for folders do not show the correct icon. Fixing this bug has been deferred until after DR1.

The spinning-arrows updating icon is not shown when opening windows. Fixing this has been deferred until after DR1.

Clicking a disclosure triangle to collapse a folder's outline view may beep and then do nothing. There was a problem with the asynchronous updating of outline view which could cause the machine to crash if you closed the outline while the folder was still being synched. The fix for that problem is scheduled for post-DR1. In the meantime, this will prevent the problem. To collapse an outline, you should wait until the window has finished updating, then click the triangle. Because the Finder also re-syncs at idle, you may have to click several times. This usually should only occur with folders with many contained items.

Both the Mac OS 8 and Non-Mac OS 8 system folder have the same icon. Fixing this has been deferred until after DR1. We also need some strong direction from HI for suggested icons if we actually do show both folders.

Editing the name of an icon on the desktop may be difficult, because the name will be painted over the desktop pattern. This is a known problem, and will be fixed post-DR1.

The Finder does not reopen windows on startup if they were left open when shutdown. This will not be fixed until after DR1.

The text of a file name is incorrectly right justified while editing file names in the Finder. This is caused by TextEdit bug 1349248 and will be corrected when the bug is fixed. Until that time

editing works but is visually peculiar.

•Folder Manager

Dependencies: MacOS Loader

Changes:

Properly handles volumes not containing a blessed folder (CD-ROMs, floppies) (1335952); fixed problems involving attributes of newly created folders and internal error propagation. (1354074)

Known limitations: The current version does use the MacOS Loader's FolderDescriptor resources but does not use the MacOS Loader's folder routing resources. In particular, the trash folder for File Sharing is not supported, and the system disk is assumed to be vRefNum - 1.

•Graphics

The graphics groups includes five subgroups: Graphics, Printing, Type and ColorSync

Graphics

Things currently in the system:

- * ColorPicker
- * QuickDraw
- * GX Graphics
- * Monitors Control Panel
- * Icon Utilities
- * Display Manager

Things yet to come:

- * Sound & Displays (replaces Monitors)
- * Full anti-aliased text support (partial implementation at this point, only for QuickDraw)

Things known not to work:

- * PICTUtilities still mostly unimplemented.

Things recently working in this build:

- * PICT recording and playback now believed to be correct.
- * Monitors should be working reasonably well now.
- * StretchBits now fully implemented, probably does have some bugs still.

Printing

Things currently in the system:

- * LaserWriter printer driver
- * ImageWriter printer driver
- * ImageWriter LQ printer driver

- * CustomWriter GX printer driver (creates PICT files)
- * Printing extensions
- * PrinterShare GX
- * Printing Finder Extension (desktop printers, Printing menu)
- * Network printing requires installation of QuickDraw GX OT Support printing extension (comes installed).
- * The Chooser can now be used to select a printer. NOTE: when you hit the create button you'll see a "Printer Setup" dialog. HIT CANCEL AT THIS DIALOG OR YOU WILL CRASH!!!!

Things yet to come:

- * Printing to serial and non-networked PostScript devices.
- * Full PFE Functionality, including:
 - Shared printers (post-DR1 deliverable)
 - Input Trays... (Printing menu)
 - Extension Setup... (Printing menu)
 - Dragging job to/from "active" area of DTP window

Things known not to work:

- * Selecting "Input Trays" from desktop printer "Printing" menu causes a crash unless:
 1. You're using the LaserWriter GX driver
 2. You're printing to an Apple printer
 3. You've previously printed to the printer you're configuring the input trays for. Workaround: Do 1, 2, & 3 above.
- * Selecting "Disable Manual Feed Alerts" from the desktop printer "Printing" menu causes a crash.
- * 1351752 - Dropping printing extensions into the Extensions folder doesn't enable them. There's a bleeding edge fix about to be released for this.
- * Cosmetic problems with print dialogs
- * Putting a spool file or PDD in the root window of your Mac OS 8 boot drive will generate a lot of assertions and an access fault.
- * Embedded fonts in PDDs and spool files will not work for DR1.
- * Non-PAP (ie Serial) connections do not work because of OT compatibility issues. See #1336499.
- * Putting a printing extension on your desktop will cause a benign "IsExtensionInvisible" assertion to fire.
- * No performance work has been done yet. Keep that in mind as you bring up dialogs. 8P

Things recently working in this build:

- * You no longer need to reboot between spooling and when PrinterShare GX is launched by the Finder
- * Printing to networked LaserWriters works from GX-aware and non-GX apps
- * Printing to CustomWriter GX (PICT creating printer driver), and PDDMaker works (this was actually true in d11e3)
- * Annoying assertion fixes (1352978, 1343997, 1349107, 1349104, 1343284)

- * Renaming desktop printers behaves as expected
- * Any 68k applications which append things to the old dialogs with AppendDITL should work now.
- * 1348931 - PrinterShare GX 'appe' (background app) works again

Type

Things currently in the system:

- * TrueType 3 Font Scaler Server
 - Supports Anti-aliased rendering
 - Supports Morisawa fonts
- * Adobe Type 1 Font Scaler Server
 - Supports Anti-aliased rendering
 - Supports Multiple Master fonts
 - Requires "enabled" Type 1 fonts (i.e. repackaged in 'sfnt's)
- * QuickDrawText
 - Supports double byte systems directly, including proportional Kanji
 - Supports anti-aliased text
 - New feature: TextStrike (already heavily used by TextObjects)
- * Font Object System
- * Glyph Cache
- * Layout Core
- * Open Font Architecture 2

Things yet to come:

- * Kerning
- * Support for linked fonts ('kids' table)
- * Font file activation support
- * NFNTs (bitmap fonts) support for GX

Things known not to work:

- * QDSetStrikeFeatures() is not supported yet
- * QDSetStrikeVariations() is not supported yet
- * QDSetStrikeMapping() is not supported yet
- * GXNewFont(), GXDisposeFont() and font editing routines are not yet implemented.
- * The routines from ScriptLayout.h involving text drawing/measuring are not yet incorporated in QuickDrawText (a DR2 deliverable).
- * Complex 1-byte scripts (specifically Arabic) are not supported yet (a DR2 deliverable).
- * Kerning functionality in GX is not operational, text simply is not kerned.
- * Live font suitcase manipulation, including adding/removing and editing suitcases. To make a new suitcase active requires rebooting after dragging the suitcase to the Fonts Folder.
- * Deep bitmap fonts (color NFNTs) are currently not supported.
- * For sfnt fonts, the "widMax" field in FontInfo does not necessarily return the same values as in 7.5; the OFA interface does not provide the data. (We suspect the 7.5 values were not correct either).
- * Certain sfnt fonts return ascent and descent values different from what the previous TrueType

scaler provided. It

appears that the previous scaler was wrong.

* FMLowMemSubstitutes and LowMem switch list are not in sync. There should be only one system-wide global

WidthTableList (not per-context).

* Anti-aliased text blitting doesn't work in 32-bit pixeldepth.

Things recently working in this build:

* Fonts embedded in applications; FONT-only fonts

* Shadowing of FOND resources (backwards compatibility: allows searching resource chain for fonts to work) Vertical

text (flag bit in TextStrike options)

* Chicago can be used as font (familyID = systemFont conflict resolved)

Bugs Fixed:

1302307 RealFont(0,-1)=true, RealFont(2,0)=false, incorrect values returned

1338100 Chicago id=0 conflicts with systemFont selector

1339589 Crash while launching Adobe PageMill 1.0.2 in MemAllocators

1341763 Access Fault on selecting A Menu in Freehand 5.5

1343187 QL Choosing get info for system file produces Access Fault

1343593 QDSetStrikeElements() does not handle kPreserveTextStrikeElement params

properly

1343727 Excess Assertions in Glyphs Manager

1343819 QD crashes while recording icon drawing (copy icon using finder get info)

1343826 FindTextStrikeTag causes Access Fault when drawing a string

1343848 DrawString crashed while calling FindTextStrikeTag

1344313 Type 1 Scaler doesn't clear variation count in QD metrics call

1345559 Changing Size from Font Menu in SimpleText causes font to revert to Geneva

1346358 char widths are wrong with some 1-byte chars (high ascii) in J fonts

1346593 SquishBitmap may spill over

1347275 draw garbage (random img) for space character in Japanese fonts

1348221 Character heights are different on System 7.5 and Mac OS 8 (e.g. Monaco 9pt bold)

1348664 RealFont() crashing for "Tekton Plus" font

1349076 plain 2-byte char text in Osaka font is condensed

1349744 GetFontInfo results bad within OpenPicture

1351104 A system font cannot be changed unless the machine is restarted.

1351757 ArchiCAD launch fails at DrawJustified

1351933 systemFont, applFont, LMGetxxFontxx() return wrong font IDs on CH sys

1351984 Deblessing & blessing Mac OS Folder causes files/folder to lose titles

1353670 Extraneous assert in GenerateFontResult

ColorSync

Things currently in the system:

* ColorSync shared library which contains:

- The 'ColorSync' pef - now contains routines for colorspace conversion as part of the ColorSync 2.1 API.

- The default Apple/Lino cmm component (type 'cmm ').

- The colorspace conversions component (type 'ccnv').
- The 'ccnv' component is only included for backwards compatibility.
- Bundle resource, icons, etc.

Things yet to come:

- * ColorSync control panel application.
- * Implementation of the CMM as a shared library.

Things known not to work:

- * CWMatchBitmap may not work for some new bitmap colorspaces

• High-Level Toolbox (HLTB) and Runtime

Notes for the HLTB in general, as well as component specific areas of HLTB, HIOObjects, and Runtime.

I. General HLTB Notes

The HIOObjects APIs are definitely going to change after DR1. The remaining HLTB work planned for D11E5 is limited to bug fixes, primarily for compatibility in the toolbox managers and the trap table, as well as continued bug fixing in HIOObjects. I.E. Qualification Holes (Unit Test Coverage and Code Coverage).

II. HLTB Component Specific Notes

II.A. Clipboard Manager - Functionally complete. II.B. Collections - Functionally complete. See detailed section below. II.C. Control Manager - Functionally complete. II.D. Core Controls - Functionally complete. II.E. Dialog Manager - Functionally complete. II.F. Drag Manager - Functionally complete. II.G. Edition Manager - The Edition Manager is functionally complete but the expectation in DR1 is that only a subset of the API will be used by clients, and this is the subset we will focus our qualification efforts on.. II.H. Gestalt - Functionally complete. II.I. Hash Table - Functionally complete. II.J. Help Manager - The complete implementation of the Balloon Help APIs is not planned for DR1. In DR1 we will essentially stub out Balloons entirely. This is done cleanly, in such a way that applications will not be adversely affected. HMIIsBalloon will return false, and all routines except HMGetHelpMenuHandle will return hmNotImplErr. HMGetHelpMenuHandle will return the Help Menu handle. See bugs 1332857, 1332446, and 1306873 for additional details. II.K. Interface Lib - Functionally complete. II.L. List Manager - Functionally complete. II.M. Menu Manager - Functionally complete. See section below. II.N. Notification Manager - - D11E3: The implementation of the Notification Manager in D11E3 is a simple non-functional stub, to only ensure that the calls return and do not crash. We are investigating adding some implementation for DR1 which will support the notification call back procs if/as required to enable compatibility applications to run. See bugs 1294111 and 1344838 for additional details. - D11E4: New for D11E4, we've fixed bug 1344838 and now invoke the installed callback procs with context. II.O. Package Manager - Functionally complete. II.P. All clients of orphan APIs must move onto the HIOObjects APIs asap after DR1. II.Q. Scrap Manager - Functionally complete. II.R. Standard Theme - The HLTB implementation of Truth Theme in code has been replaced by the MuSE Data Driven Truth Theme in D11. II.S. Toolbox

Utilities - Functionally complete. II.T. Translation Manager - This feature is not planned until DR2. II.U. Window Manager - Functionally complete.

III.Menu Manager Specific Notes

Classic Menu Manager, as described in Inside Macintosh, (b) the "Orphaned" Menu Manager, as described in the original Mac OS 8 Menu Manager ERS, and (c) the New HIMenus class, a SOM-based implementation.

• **Installer Script**

"Not enough capacity" error message now reflects minimum HD capacity value of 230MB for DR1. No longer call buggy equalstring. Partition flavors should be correctly determined now. Script will recognize ATA driver.

• **I/O**

The release notes are broken into 2 sections, one for the I/O system in general, and one for the specific components.

This does not cover networking/OpenTransport Family and plugins.

General I/O system for DR1

There are some general facilities not fully implemented in the DR1 I/O system. These areas include: Arbitration Services, Power Management, Cancellation, Hot plugging of devices (where HW supports), and Hot plugging of new software (families and Plug-ins). For all of the bus families there will be new requirements for OF support and/or enablers to both describe multifunction cards and to provide supapeture information.Device to Device will require some work in all the families.

I/O components will be modified to support these facilities in the future. Some interfaces will be modified, added, or deleted. Attempts will be made to minimize the impact to the developer.

Specific Components

Bootimg - Does not support 3rd party chaining during boot process, such as volume encryption or volume password.

Device Manager - Applications cannot access Native FPI's directly.

Input Devices - Areas not supported are: Acceleration Cursor Device Manager Compatibility

Tablet Spec compatibility

Persistent mode settings Device switching between absolute and relative mode

NuBus - Slot Manager API's not yet supported

PCI - Address space limitations

Sounds - Sound in not supported

Keyboard Family - eliminated in an earlier build, Keyboard I/O functionality now supported by the User-Input Family

• Mac OS 8 StdCLib

The Standard C Library is documented in the following two standards:

1. ANSI/ISO Standard 9899-1990 American National Standard for Programming Language - C
2. ISO/IEC 9945-1:1990, IEEE Std 1003.1-1990 Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]

The Standard C Library for Mac OS 8 is divided into three files:

StdCLib - The code fragment library StdCLib is maintained for legacy code and is available only in the

Classic Mac OS System 7 or before environment.

CLibraryStandard - the ANSI standard and some parts of the POSIX standard.

CLibraryAppleExtra - the Apple extensions.

The following functions have not been determined to be thread safe:

errno

close

creat

dup

fcntl

lseek

open

read

unlink

write

setlocale

localeconv

raise

signal

remove

rename

tmpfile

tmpnam

fclose

fflush

fopen

freopen

setbuf

setvbuf

fprintf

			fscanf	
sprintf	printf	scanf		
			sscanf	
vsprintf	vfprintf	vprintf		
		fgetc	fgets	
fputs	fputc			
			getc	getchar
putc	gets			
			putchar	
puts	ungetc			
fread			fwrite	
	fgetpos	fseek		
fsetpos			ftell	rewind
	clearerr			
feof			ferror	
perror		fdopen		
rand			srand	
abort		atexit		
exit			getenv	
MoneyData		NumericData		
TimeData				
	getIDstring	mktemp		
access				
ioctl			faccess	
	MacOSErr	IEResolvePath		
fsetfileinfo				
getw		putw		

The following functions have not been implemented:

- MakeResolvedFSSpec
- ResolvePath
- MakeResolvedPath
- ResolveFolderAliases

Known bugs and limitations are:

There is no implementation of the concept of working directory.

Need to increase size of "jmp_buf" for 64-bit integer registers on PPC-620

RN: Dev:Console cannot be opened with read-write access StdCLib 3.4
StdCLib stdio fopen() "w" option opens files for append rather than write.

SdtCLib stdio fopen() created files have incorrect file creator/type
setlocale is not fully implemented.

setvbuf doesn't use all of buffer given StdCLib Mac OS 8

unlink() should return -1 if there are any open file descriptors to the file

fopen returns invalid FILE* after creating file in the current folder

Remove fails to set errno when called to remove an object that does not exist

open("foo", (RDWR | O_CREAT | O_EXCL)) fails even though file doesn't exist

lseek always fails when passed a negative offset

fopen(,"r") should allow multiple FILE pointers to read from the same file

• **MicroKernel**

The kernel is functionally complete. Error codes will change post-DR1.

Known problems not covered by bugs: 1. ReplyToMessage at SIH level may fail randomly. We've never seen this happen; there may be no clients. 2. The kernel needs additional stress testing. Exhausting resources (e.g. creating tasks until you can't create any more) may result in a crash. 3. Heavy paging demand may cause an "unable to get free page" failure; this is a joint problem with the kernel and backing provider to be solved by ongoing tuning.

• **Preferences Manager**

Creates a resource fork as necessary for existing prefs files that lack one.

• **Process Manager**

Changes since DR0 - Finder launched by type and creator ('ÄNDR', 'MACS') Any app used as a "replacement" finder must have this signature - Cooperative interrupt qtypes must be "known" via OSUtils.h, or created via CreateCooperativeInterruptType - ProcessesPreemptive library supports preemptive clients for a subset SPI/API - Single machine debugging support - see ProcessesDebugPriv.i - Toolbox critical section support for abnormal termination

Known Problems - Single machine debugging support requires debugger to distinguish between preemptive and cooperative debug targets. Also, debugger is required to know whether process memory address range is in lowmem or not. In addition, there are three known bugs:
1343256 Single machine debugger deadlocks in Processes calls if task-created exception has not yet been handled

1346296 StopCurrentProcess does not work before target has called InitWindows
1349414 If an extension ('appe') quits before the Finder is launched, the System Process attempts to launch the Finder twice.

•System Logging Notes

The main API routine, LogSituation, is functional and complete.
The administration API side of System Logging is still in flux. This should only concern MuSE and the Diagnostic folks.
Usage guidelines are being developed and probably will not be available until post DR1.

•Text and International Engineering

What's New In This Build:

No functional changes. KeyTranslationPrototypes.h changes occurred in this build. This tracks changes interfaces which eliminated the LowLevelKeyEvent type.

Known problems:

D11E4 StubLibrary for KeyTranslation builds a PreemptiveSafe version of the stub. This is incorrect, KeyTranslation imports from TextInputSystem which is dependent on cooperative services. Do not link against KeyTranslation in preemptive code.

© 1996 Apple Computer, Inc. All rights reserved.

Apple, the Apple logo, and Macintosh are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. Mac, Mac OS, and Mac OS 8 are trademarks of Apple Computer, Inc.