THE PROGRAMMER'S APPLE® MAC® SOURCEBOOK

Reference Tables for Apple Macintosh® Hardware and System Software

Thom Hogan

Microsoft PRESS

THE PROGRAMMER'S APPLE® MAC® SOURCEBOOK

Reference Tables for Apple Macintosh® Hardware and System Software

Thom Hogan

THE PROGRAMMER'S APPLE® MAC® SOURCEBOOK

Reference Tables for Apple Macintosh* Hardware and System Software

Thom Hogan

Microsoft PRESS®

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
16011 NE 36th Way, Box 97017, Redmond, Washington 98073-9717

Copyright © 1989 by Thom Hogan
All rights reserved. No part of the contents of this book may
be reproduced or transmitted in any form or by any means without
the written permission of the publisher.

Library of Congress Cataloging in Publication Data
Hogan, Thom, 1952–
The programmer's Apple Mac sourcebook / Thom Hogan.
p. cm.
Bibliography: p.
Includes index.
ISBN 1-55615-168-3 : $22.95
1. Macintosh (Computer)--Programming. I. Title.
QA76.8.M3H63 1989
005.265--dc19 89-3195
CIP

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 WAKWAK 3 2 1 0 9

Distributed to the book trade in the
United States by Harper & Row.

Distributed to the book trade in
Canada by General Publishing Company, Ltd.

Distributed to the book trade outside the
United States and Canada by Penguin Books Ltd.

Penguin Books Ltd., Harmondsworth, Middlesex, England
Penguin Books Australia Ltd., Ringwood, Victoria, Australia
Penguin Books N.Z. Ltd., 182–190 Wairau Road, Auckland 10, New Zealand

British Cataloging in Publication Data available

Apple® AppleTalk® HyperCard® ImageWriter® LaserWriter® Lisa® Mac® and
Macintosh® are registered trademarks, and Apple Desktop Bus™ Finder™ MPW™
MultiFinder™ QuickDraw™ and Switcher™ are trademarks of Apple Computer, Inc.
IBM® is a registered trademark of International Business Machines Corporation.

Project Editor: Mary Ann Jones Technical Editors: Adrian Falla, Tom Creedon

To Andy Hertzfeld, Bill Atkinson, Burrell Smith, and a host of others who helped make the Mac what it is—but they probably know all this stuff already, anyway.

Contents

Preface xxi
Acknowledgments xxiii
Introduction xxv

SECTION I: GENERAL INFORMATION 1

Numeric Conversions
Hexadecimal
1.01. Hexadecimal to Decimal Number Conversion 3
1.02. Hexadecimal to Binary Number Conversion 4
1.03. Hexadecimal Addition Tables 5
1.04. Hexadecimal Multiplication Tables 7
Binary
1.05. Binary Number Conversions 8
1.06. Binary to Signed Decimal Number Conversion 9
Decimal
1.07. Decimal to Binary Number Conversion 10
1.08. Decimal to Hexadecimal Number Conversion 11
Data Formats
1.09. Two's Complements 12
1.10. Common 680x0 Data Formats 13
1.11. Common Macintosh Data Formats 14
1.12. Common String Formats 16
1.13. Common Memory Area Terminology 16
1.14. Binary Coded Decimal Number Format 17
Character Sets
1.15. ASCII Control Codes 18
1.16. ASCII Character Set 19
1.17. Apple Macintosh Character Set 20
Other
1.18. Digit Positions in Common Bases 22
1.19. Powers of Two 22
1.20. International Sort Ordering 23
1.21. Truth Tables for Logical Operations 25

SECTION II: OPERATING SYSTEM SOFTWARE, RESOURCES, AND FILE FORMATS 27

Resources
Standard and Predefined Resources
2.001. Predefined System Packages 29
2.002. Predefined Cursors 29
2.003. Standard Package IDs 30
2.004. Standard Drivers and DAs 30
2.005. Standard Patches (PTCHs) 30
2.006. Standard Alert Icons 31
2.007. ROM Resources 31
2.008. Resource Numbering 32

2.009. Resource Attributes 33
2.010. Resource Types 33
2.011. actb, dctb, ictb Resource Formats 35
2.012. ALRT Resource Format 35
2.013. BNDL Resource Format 36
2.014. cctb Resource Format 36
2.015. CDEF Resource Format 37
2.016. cdev Resource Format 37
2.017. cdev Messsage Values 37
2.018. cdev Result Codes 38
2.019. cicn Resource Format 38
2.020. clut Resource Format 39
2.021. CNTL Resource Format 39
2.022. CODE Resource Format 39
2.023. crsr Resource Format 40
2.024. CURS Resource Format 40
2.025. Cursor Data Masks 41
2.026. DITL Resource Format 41
2.027. DLOG Resource Format 42
2.028. fctb Resource Format 42
2.029. FKEY Resource Format 42
2.030. FOND Resource Format 43
2.031. FONT Resource Format 43
2.032. FREF Resource Format 44
2.033. ICON Resource Format 44
2.034. ICN# Resource Format 44
2.035. INIT Resource Format 44
International Resource Formats
2.036. INTL 0 or itl0 Resource Format 45
2.037. INTL 1 or itl1 Resource Format 46
2.038. itlb Resource Format 46
2.039. KCHR Resource Format 47
2.040. KMAP Resource Format 47
2.041. mach Resource Format 48
2.042. MBAR Resource Format 48
2.043. MBDF Resource Format 48
2.044. mctb Resource Format 49
2.045. MDEF Resource Format 49
2.046. MENU Resource Format 49
2.047. NFNT Resource Format 50
2.048. nrct Resource Format 50
2.049. PAT Resource Format 50
2.050. PAT# Resource Format 51
2.051. PICT Resource Format 51
2.052. pltt Resource Format 51
2.053. ppat Resource Format 52
2.054. PRER Resource File Format 52
2.055. PRES Resource File Format 53
2.056. ROv# Resource Format 53

2.057. scrn Resource Format 53
2.058. SIZE Resource Format 54
2.059. Format 1 snd Resource Format 54
2.060. Format 2 snd Resource Format 54
2.061. snth Resource Format 55
2.062. STR Resource Format 55
2.063. STR# Resource Format 55
2.064. TEXT Resource Format 55
2.065. vers Resource Format 56
2.066. wctb Resource Format 56
2.067. WDEF Resource Format 57
2.068. WIND Resource Format 57
2.069. Resource File Format 58
2.070. RMAKER Syntax 59
2.071. REZ Compiler Formats 60
2.072. Font ID Numbers 66
2.073. SysEnvirons Results 74

User Interface

2.074. Predefined Keys 75
2.075. Interface components 76
2.076. Common Icons 76

System and Finder

2.077. System Software History 77
2.078. Files Supplied with System 78
2.079. File Creators and Types 80
2.080. System Error Messages 88
2.081. System Software Result Codes 89

Disk and File Formats

2.082 Macintosh Disk Format 90
2.083. Microfloppy Sectors per Track 91
2.084. MacBinary File Format 91
2.085. MacPaint File Format 91
2.086. PackBits File Format 92
2.087. PICT File Format 92
2.088. PicComments 92
2.089. PICT Data Types 93
2.090. PICT Opcodes 94
2.091. TIFF File Format 97
2.092. MFS Disk Format 100
2.093. MFS Volume Information 100
2.094. MFS Volume Allocation Map 101
2.095. MFS File Directory Format 101
2.096. HFS Disk Format 101
2.097. HFS Volume Information 102
2.098. HFS Boot Blocks Layout 103
2.099. B*-Tree Node Record Layout 103
2.100. Extents Key Record Layout 104
2.101. Catalog Key Record Layout 104
2.102. Directory Record Layout 105

2.103. File Record Layout 105
2.104. Thread Record Layout 106
Installer
2.105. Installer Script Format 107
2.106. Installer File Spec Flags 108
2.107. Installer Resource Spec Flags 108
HyperCard
2.108. HyperTalk Commands 109
2.109. HyperTalk Properties 112
2.110. HyperTalk Constants 113
2.111. HyperTalk Messages 113
MPW
2.112. MPW Commands 115
2.113. MPW Command Status Values 117
2.114. MPW Command Terminators 117
2.115. Special Characters in MPW Commands 118
2.116. Expressions in MPW Commands 119
2.117. Editing Selection Operators in MPW Commands 119
2.118. MPW Tools 120
2.119. MPW MacsBug Exceptions 121
2.120. MPW MacsBug Commands 122
2.121. MPW Object File Format 123

SECTION III: ROM TRAPS AND SYSTEM CALLS 125

3.001. ROM Versions 127
System Traps
3.002. Trap Word Format 128
3.003. System Traps by Name 129
3.004. System Traps by Number 144
3.005. System Traps by Manager 160
3.006. System Traps and Routines That May Move or Purge Memory 175
3.007. Global Variables 189
Toolbox Managers
3.008. Interface Toolbox Managers 192
Resource Manager
3.009. Resource Manager Routines 193
3.010. Resource Manager Result Codes 194
QuickDraw
3.011. QuickDraw Regions/Pixels/Pictures Routines 195
3.012. Miscellaneous QuickDraw Routines 197
3.013. QuickDraw GrafPort Routines 197
3.014. QuickDraw Pen/Line Drawing Routines 198
3.015. QuickDraw Cursor Routines 198
3.016. QuickDraw Text Drawing Routines 199
Constants and Data Structures
3.017. Constants 200
3.018. Cursor Data Structure 201
3.019. QuickDraw GrafPort Data Structure 202

3.020. QuickDraw Text Style (StyleItem) Equates 202
3.021. QuickDraw Point Data Structure 203
3.022. QuickDraw Rect Data Structure 203
3.023. QuickDraw Region Data Structure 203
3.024. QuickDraw Bitmap Data Structure 204
3.025. QuickDraw Color Equates (Old Style) 204
3.026. QuickDraw Picture Data Structure 205
3.027. QuickDraw Polygon Data Structure 205
3.028. QuickDraw Pen State Data Structure 205
3.029. QuickDraw Procs Data Structure 206
3.030. QuickDraw FontInfo Data Structure 206
3.031. QuickDraw Pen Transfer Modes 206
3.032. QuickDraw Coordinate Plane 207
3.033. QuickDraw InitGraf Default Settings 207

Color QuickDraw

3.034. Color QuickDraw Routines 208
3.035. RGBColor Data Structure 209
3.036. Color Table Data Structure 209
3.037. ColorSpec Data Structure 209
3.038. ColorGrafPort Data Structure 210
3.039. PixMap Data Structure 211
3.040. PixPat Data Structure 211
3.041. Color QuickDraw Transfer Modes 212
3.042. Color QuickDraw Arithmetic Drawing Modes 212
3.043. Color QuickDraw Cursor Data Structure 213
3.044. Color QuickDraw Icon Data Structure 213
3.045. GrafVars Data Structure 213
3.046. RGB Equivalents of Old QuickDraw Colors 214
3.047. QuickDraw Additive Color Table 214

Color Manager

3.048. Color Manager Routines 215
3.049. ReqListRec Data Structure 216
3.050. SProcRec Data Structure 216
3.051. CProcRec Data Structure 216

Font Manager

3.052. Font Manager Routines 217
3.053. Font Numbers 218
3.054. FontRec Data Structure 218
3.055. FMInput Data Structure 219
3.056. FMOutput Data Structure 219
3.057. Font Characterization Table 220
3.058. Font Metrics Data Structure 220
3.059. FamRec Data Structure 221
3.060. WidthTable Data Structure 222

Script Manager

3.061. Script Manager Routines 223
3.062. Script ID Numbers 224
3.063. CharByte and CharType Values 225

3.064. Transliterate Target Values 225
3.065. GetScript Verb Parameter Values 226
3.066. GetEnvirons Verb Parameter Values 226
Toolbox Event Manager
3.067. Toolbox Event Manager Routines 227
3.068. Toolbox Event Priorities 227
3.069. Toolbox EventRecord Data Structure 228
3.070. Toolbox Event Codes 228
3.071. Toolbox Event Masks 229
3.072. Toolbox Event Modifier Flags 229
Window Manager
3.073. Window Manager Routines 230
3.074. Window Types 232
3.075. WindowRecord Data Structure 232
3.076. CWindowRecord Data Structure 233
3.077. AuxWinRec Data Structure 233
3.078. WinCTab Data Structure 234
3.079. Mouse Locations in Windows 234
3.080. Window Part Identifiers for Color Tables 234
3.081. Window Definition Messages 235
3.082. Window Hit Messages 235
Control Manager
3.083. Control Manager Routines 236
3.084. Control Types 237
3.085. Control Part Codes 237
3.086. ControlRecord Data Structure 237
3.087. AuxCtlRec Data Structure 238
3.088. CtlCTab Data Structure 238
3.089. Control Part Identifiers for Color Tables 238
3.090. Control Axis Parameter Values 239
3.091. Control Definition Messages 239
Menu Manager
3.092. Menu Manager Routines 240
3.093. MenuInfo Data Structure 242
3.094. MCEntry Data Structure 242
3.095. Menu Color Information Table 243
3.096. MenuCRsrc Data Structure 243
3.097. Menu List Data Structure 243
3.098. MenuSelect ID/Item Return 243
3.099. Menu Definition Messages 244
3.100. Menu Definition Metacharacters 244
3.101. Menu Bar defProc Messages 244
TextEdit
3.102. TextEdit Routines 245
3.103. TERec Data Structure 246
3.104. TextEdit Justification Codes 246
3.105. TEStyleRec Data Structure 247
3.106. STElement/LHElement Data Structures 247

3.107. NullSTRec Data Structure 247
3.108. TextStyle Data Structure 248
3.109. StScrpRec Data Structure 248
3.110. ScrpSTElement Data Structure 248
3.111. SetStyle Attribute Controls 248
Dialog Manager
3.112. Dialog Manager Routines 249
3.113. Dialog Item Types 250
3.114. Dialog Item Lists 250
3.115. Dialog Color Table 251
3.116. Dialog Standard Item Constants 251
3.117. DialogRecord Data Structure 251
3.118. DialogTemplate Data Structure 252
3.119. AlertTemplate Data Structure 252
3.120. Alert StageList Data Structure 252
Desk Manager
3.121. Desk Manager Routines 253
3.122. editCmd Parameters 254
3.123. DA Device Driver Format 254
3.124. DA Device Driver Control Routine Messages 254
Scrap Manager
3.125. Scrap Manager Routines 255
3.126. ScrapStuff Data Structure 256
3.127. Scrap Data Formats 256
3.128. Scrap Manager Result Codes 256
Toolbox Utilities
3.129. Toolbox Utility Routines 257
3.130. FixedAndInt Data Structure 258
3.131. Int64Bit Data Structure 258
3.132. SysEnvRec Data Structure 258
3.133. SysEnv Result Codes 259
Package Manager
3.134. Package Manager Routines 259
3.135. Supplied PACKs 260
3.136. Binary-Decimal Package Routines 260
3.137. International Package Routines 261
3.138. Int10Rec Data Structure 262
3.139. Intl1Rec Data Structure 263
3.140. International Routine Selectors 263
3.141. International String Comparison Result Codes 263
3.142. Standard File Package Routines 264
3.143. SFReply Data Structure 265
3.144. Standard File Routine Selectors 265
3.145. Standard File Predefined Constants 266
3.146. Standard File Resource IDs 266
Palette Manager
3.147. Palette Manager Routines 267
3.148. Palette Manager Color Methods 268
3.149. ColorInfo Data Structure 268

3.150. Palette Data Structure 268
List Manager
3.151. List Manager Routines 269
3.152. ListRec Data Structure 270
3.153. List Selection Flag Byte 270
3.154. List Manager Masks 271
3.155. List Manager Messages 271
Operating System Managers
3.156. Operating System Calls 272
Memory Manager
3.157. Memory Manager Routines 273
3.158. Memory Organization for 128/512/Plus 274
3.159. Memory Organization for XL 275
3.160. Memory Organization for SE 275
3.161. Memory Organization for II 276
3.162. Zone Data Structure 276
3.163. Zone Block Header Data Structure 277
3.164. Tag Byte Format 277
3.165. Master Pointer Structure 277
3.166. InitZone Parameter Block 278
3.167. Memory Manager Result Codes 278
Segment Loader
3.168. Segment Loader Routines 279
3.169. Finder Information Data Structure 279
3.170. AppFile Data Structure 280
3.171. Jump Table Formats 280
Event Manager
3.172. Event Manager Routines 281
3.173. EventRecord Data Structure 282
3.174. EvQEl Data Structure 282
3.175. Event Codes 283
3.176. Keyboard Event Message 283
3.177. KeyTrans Data Structure 283
3.178. Event Masks 284
3.179. Event Modifier Flags 284
3.180. PostEvent Result Codes 284
File Manager
3.181. File Manager Routines 285
3.182. FInfo Data Structure 286
3.183. FXInfo Data Structure 286
3.184. DInfo Data Structure 286
3.185. DXInfo Data Structure 287
3.186. File Manager Result Codes 287
3.187. File Positioning Modes 288
3.188. File Information Flags 288
3.189. File Access Modes 289
3.190. File Access Permissions 289
3.191. ParamBlockRec Data Structure 290
3.192. HParamBlockRec Data Structure 291

3.193. CInfoPBRec Data Structure 292
3.194. CMovePBRec Data Structure 293
3.195. WDPBRec Data Structure 293
3.196. FCBPBRec Data Structure 294
3.197. Volume Control Blocks Data Structure 294
3.198. File Control Block Data Structure 296
3.199. Drive Queue (DrvQEl) Data Structure 296
Device Manager
3.200. Device Manager Routines 297
3.201. Device Manager Result Codes 298
3.202. Device ParamBlockRec Data Structure 298
3.203. Device Driver Structure 299
3.204. Device Driver Flags 299
3.205. DCtlEntry Data Structure 300
3.206. I/O Queue Record Data Structure 300
3.207. Device Unit Table 301
3.208. Device Access Modes 301
3.209. Standard Unit Assignments 301
3.210. Chooser Device ID Flags 302
3.211. Device Positioning Modes 302
Interrupts
3.212. Interrupt Priority Levels 302
3.213. Processor Priority Levels 303
3.214. Primary Interrupt Vector Table 303
3.215. Level 1 Secondary Interrupt Vector Table 303
3.216. Level 2 Secondary Interrupt Vector Table 304
3.217. Level 2 External/Status Interrupt Vector Table 304
Graphics Devices
3.218. Graphics Devices Routines 305
3.219. gDevice Data Structure 306
3.220. Graphics Devices Attribute Types 307
Sound Driver
3.221. Sound Driver Routines 307
Square Wave Synthesizer
3.222. Sound Driver Synthesizer Types 308
3.223. SWSynthRec Data Structure 308
3.224. Note Values 308
Four-Tone Synthesizer
3.225. Tone Data Structure 313
3.226. FTSynthRec Data Structure 313
3.227. FTSoundRec Data Structure 313
Free Form Synthesizer
3.228. FFSynthRec Data Structure 314
Disk Driver
3.229. Disk Driver Routines 314
3.230. Driver Positioning Modes 314
3.231. DrvSts Data Structure 315
3.232. Disk Driver Result Codes 315
3.233. Advanced Control Calls 316

SCSI Manager
3.234. SCSI Manager Routines 316
3.235. SCSI Operations 317
3.236. SCSI Control and Status Bits 317
3.237. SCSI Driver Descriptor Map 318
3.238. SCSI Device Partition Map 318
3.239. SCSI Result Codes 318
Sound Manager
3.240. Sound Manager Routines 319
3.241. Sound Manager Synthesizer Types 320
3.242. SndChannel Data Structure 320
3.243. ModifierStub Data Structure 320
3.244. Sound Manager Commands 321
Serial Drivers
3.245. Serial Driver Routines 321
3.246. Serial Driver Reference Numbers 322
3.247. Driver Reset Word 322
3.248. SerShk Data Structure 323
3.249. SerStaRec Data Structure 323
3.250. Advanced Control Calls 323
3.251. Serial Driver Result Codes 324
System Error Handler
3.252. System Error Handler Routine 324
3.253. System Errors 325
Operating System Utilities
3.254. Operating System Utility Routines 326
3.255. Queue Types 327
3.256. DateTimeRec Data Structure 327
3.257. Operating System Result Codes 327
Printing Manager
3.258. Printing Manager Routines 328
3.259. TPrint Data Structure 329
3.260. TPrInfo Data Structure 329
3.261. TPrJob Data Structure 329
3.262. TPrStatus Data Structure 330
3.263. Printing Manager Result Codes 330
3.264. Printing Control Constants 331
3.265. TGnlData Data Structure 331
3.266. TRslRg Data Structure 332
3.267. TRslRec Data Structure 332
3.268. TSetRslBlk Data Structure 332
3.269. TDftBitsBlk Data Structure 333
3.270. TGetRotnBlk Data Structure 333
Slot Manager (NuBus)
3.271. Slot Manager Routines 334
3.272. SExecBlock Data Structure 337
3.273. SpBlock Data Structure 337
3.274. SInfoRecord Data Structure 338
3.275. SResource Directory Data Structure 338

3.276. SResource Apple-defined IDs 339
3.277. SPRAMRecord Data Structure 339
3.278. FHeaderRec Data Structure 339
3.279. VendorInfo List IDs 340
3.280. Slot Manager Result Codes 340
Start Manager
3.281. Start Manager Routines 341
3.282. DefStartRec Data Structure 342
3.283. DefVideoRec Data Structure 342
3.284. DefOSRec Data Structure 342
Shutdown Manager
3.285. Shutdown Manager Routines 343
3.286. ShutDwnInstall Masks 344
ADB Manager
3.287. ADB Bus Routines 345
3.288. ADB Bus Command Formats 346
3.289. ADB Bus Register 3 Data Fields 346
3.290. ADBDataBlock Data Structure 346
3.291. ADBSetInfoBlock Data Structure 347
Video Driver (NuBus Card)
3.292. Video Driver Parameter IDs 347
3.293. VideoDeviceRec Data Structure 347
3.294. Video Driver Data Structures 348
Color Picker Package
3.295. Color Picker Package Routines 349
3.296. HSVColor Data Structure 350
3.297. HSLColor Data Structure 350
3.298. CMYColor Data Structure 350
Vertical Retrace Manager
3.299. Vertical Retrace Manager Routines 351
3.300. VBLTask Data Structure 352
3.301. Vertical Retrace Manager Result Codes 352
AppleTalk Manager
3.302. AppleTalk Routines 353
3.303. ABusRecord Data Structure 359
3.304. MPPParamBlock Data Structure 360
3.305. XPPParamBlock Data Structure 361
3.306. ALAP Frame 361
3.307. AddrBlock Data Structure 362
3.308. DDP Datagram Data Structure 362
3.309. ATPParamBlock Data Structure 363
3.310. ATP Packet Data Structure 364
3.311. ATP Flags Field 364
3.312. ATP Client Internet Address Data Structure 364
3.313. NBP Packet Data Structure 365
3.314. Names Table Entry Data Structure 365
3.315. EntityName Data Structure 365
3.316. RetransType Data Structure 366
3.317. AppleTalk Result Codes 366

Disk Initialization Package
3.318. Disk Initialization Routines 367
3.319. Disk Initialization Routine Selectors 367
3.320. Disk Initialization Result Codes 367
Time Manager
3.321. Time Manager Routines 368
3.322. TMTask Data Structure 368
Deferred Task Manager
3.323. Deferred Task Manager Routines 368
3.324. DeferredTask Data Structure 368
SANE
3.325. SANE Routines 369
3.326. SANE Data Types 370
3.327. SANE Not-a-Number Codes 371
3.328. Comparison Effect on 68000 Status Bits 371
3.329. Result Codes from SANE Inquiries 371
3.330. SANE Environment Word 372
3.331. Test-Exception Word 373
3.332. Exception Halt Stack Frame

SECTION IV: HARDWARE 375

4.01. Macintosh Machine Summary 377
Keyboards
Original and Mac Plus
4.02. Macintosh 128/512/Plus Keyboard Command Protocol 378
4.03. Macintosh 128/512/Plus Key Transition Codes 379
4.04. Macintosh U.S. Key Codes 380
4.05. Macintosh International Key Codes 381
4.06. Macintosh ADB Standard Key Codes 382
4.07. Macintosh ADB Extended Key Codes 383
Memory Organization
4.08. Macintosh 128/512/Plus/SE System RAM Layout 384
4.09. Macintosh 128/512/Plus/SE Video Buffer 384
4.10. Macintosh 128/512/Plus/SE Sound Buffer 385
4.11. Macintosh II System RAM Layout 385
4.12. Macintosh 128/512/Plus/SE General Memory Organization 386
4.13. Macintosh II General Memory Organization 387
4.14. Parameter RAM Layout 388
Registers and Protocols
4.15. Real Time Clock Register Commands 388
4.16. VIA Registers 389
4.17. VIA Timers 389
4.18. VIA Interrupts 390
4.19. Exception Vectors 390
Serial
4.20. SCC Data and Control Registers 391
Disk
4.21. SCSI Address Assignments 391

4.22. Disk State Control Lines and Registers 391
CPUs (680x0)
4.23. 680x0 Register Set 392
4.24. 680x0 Instruction Set 393
4.25. 680x0 Addressing Modes 396
4.26. 680x0 Exception Vectors 397
4.27. 680x0 Address Space Types 398
680x0 Stack Frames
4.28. 680x0 Normal Stack Frames 398
4.29. 680x0 Exception Stack Frames 399
4.30. 68020 Long/Short Bus Cycle Fault Stack Frames 400
4.31. 68020 Module Call Stack Frame 401
4.32. 680x0 Special Status Word 401
4.33. 680x0 Coprocessor Interface Register Set Map 402
NuBus
4.34. NuBus Memory Usage 403
4.35. NuBus Slot Memory Assignments 404
4.36. NuBus 24-Bit to 32-Bit Translations 404
4.37. NuBus to 68020 Byte Swapping 404
4.38. NuBus ByteLanes Mapping 405
4.39. NuBus Block Transfer Size and Address 405
TMx Line Use
4.40. TMx Transfer Mode Codes 406
4.41. TMx Acknowledgment Codes 406
4.42. TMx Attention Cycle Codes 407
Pinouts and Connectors
Chips and Memory
4.43. 68000 Pinouts 407
4.44. 68020 Pinouts 409
4.45. 68881 Pinouts 410
4.46. 68851 Pinouts 411
4.47. Single Inline Memory Module (SIMM Connector) 412
Buses
4.48. Apple DeskTop Bus Pinouts 413
NuBus (II)
4.49. NuBus Card Layout 413
4.50. NuBus Bus Signals (Connector) 414
SE Connector
4.51. SE Expansion Card Layout 414
4.52. SE Bus Expansion Signals (Connector) 415
LaserWriter NT
4.53. LaserWriter NTX Card Layout 415
4.54. LaserWriter NTX Bus Signals (Connector) 416
Serial
4.55. Macintosh 128/512 DB-9 Serial Port Connectors 417
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors 417
4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors 418
4.58. LaserWriter DB-9 Serial Port Connector 419
4.59. LaserWriter Mini-8 Serial Port Connector 420

4.60. LaserWriter DB-25 Serial Port Connector 420
4.61. Apple Modem Serial Port Connectors 421
Drives
4.62. Macintosh 128/512/Plus/SE External Drive Connector 422
4.63. Macintosh Plus/SE/II SCSI Connector 423
4.64 Sound Pinouts 423
Keyboard/Mouse
4.65. Macintosh 128/512/Plus Keyboard Pinouts 424
4.66. Macintosh 128/512/Plus Mouse Pinouts 424
4.67. Macintosh II Video Card Pinouts 425
Printers
4.68. ImageWriter Family History 426
ImageWriter II
4.69. ImageWriter Default Settings 427
Software
4.70. ImageWriter II Software Switches 428
4.71. ImageWriter Printer Control Functions 428
4.72. ImageWriter Character Pitch Effects on Functions 430
4.73. ImageWriter II Custom Character Width Codes 431
4.74. ImageWriter II Custom Character Data Byte 431
Hardware
4.75. ImageWriter DIP Switch Settings 432
4.76. ImageWriter Alternate Character Sets 433
LaserWriter
4.77. LaserWriter Family History 434
Software
4.78. LaserWriter NTX Disk and File-System Operators 435
PostScript
4.79. PostScript Page Types 436
4.80. PostScript Font Numbers 436
4.81. LaserWriter statusdict Persistent Parameter Operators 437
4.82. LaserWriter setsoftwareiomode Options 438
4.83. LaserWriter sethardwareiomode Options 438
4.84. LaserWriter setsccbatch Options 438
4.85. LaserWriter statusdict Volatile Parameter Operators 439
4.86. LaserWriter New PostScript Operators 439
4.87. LaserWriter Reserved Communication Codes 440
Emulation
4.88. LaserWriter NT Diablo Emulation Supported Control Characters 440
4.89. LaserWriter NT Diablo Emulation eescratch Location Assignments 441
4.90. LaserWriter NT LaserJet+ Emulation Font Mapping 441
4.91. LaserWriter NT LaserJet+ Emulation Symbol Set Differences 441
4.92. LaserWriter NT LaserJet+ Emulation Supported Control Characters 442
Hardware
Switch Settings
4.93. LaserWriter NT/NTX Operational Mode Switch Settings 446

Bibliography 447
Index 451

Preface

As Apple's prodigious paper onslaught attests, the quantity of information available about the Macintosh is already immense and increasing daily. Even with Apple's indexing efforts (*Inside Macintosh X-REF*, the *Technical Notes* cross-reference), finding the answer to a single question often requires you to consult two or three books and flip through hundreds of pages. Moreover, the constantly changing Macintosh hardware and software has made some data obsolete, modified other data, and added new data—and this information is often spread over several volumes, as well.

Two years ago, Microsoft Press and I set out to create a reference work for the IBM PC that would be reasonably complete, have excellent cross-referencing and source pointers, and serve as the first source of information for most programmers. This work is an attempt to do the same thing for Mac programmers; in essence, I've attempted to create what I call a "poor person's CD-ROM" of Macintosh information. If the information you seek isn't in this book, it is likely to be a minor detail documented in some poorly distributed work.

You will note that this is a first edition. Microsoft Press and I intend to update and expand the work as the world of Macintosh computers changes. Although I've tried to incorporate as much information as possible about new products, the Mac IIx and several other new products will get more extensive treatment in subsequent editions.

Let me know of changes and additions that you would like to see in future editions. And if you find any part of this book weaker than you think it should be, by all means let me know that, too. Although I can't promise to incorporate all suggestions, I do promise to read and seriously consider each one. Send your ideas to:

Thom Hogan
Programmer's Mac Sourcebook Suggestions
c/o Microsoft Press
16011 NE 36th Way
Box 97017
Redmond, WA 98073-9717

I hope that this work accomplishes the goal I intended to achieve and that you find it a useful compendium—a desktop reference that serves as your first stop for all information about Macintosh computers.

Acknowledgments

Many people helped make this book what it is. Susan Lammers provided the initial enthusiasm and worked on the development of the Macintosh edition before moving on to other jobs at Microsoft. Pat and Bob Combs also provided help in the initial planning and development of the project. David Rygmyr and Mary Ann Jones provided ongoing support (as well as occasional nudges) and shepherded the project through the publishing process in Redmond. Adrian Falla and Tom Creedon did a thorough and useful technical edit, and Mark Dodge reformatted and corrected the tables. As usual, the whole crew at Microsoft Press was helpful beyond expectations. I again salute them for the fine author-publisher relationship they have shaped, and I look forward to working with them in the future.

There are others to thank, as well. The mostly anonymous souls at Apple who provided the base of information in *Inside Macintosh* and the *Technical Notes* deserve a round of applause. Much of the early Apple writing is the work of Caroline Rose and Chris Espinosa, who clearly had their work cut out for them. While I found lots of minor style descrepancies in the source material, it was rare that I found an outright error.

Finally, thanks to Kurt Matthies, Cheryl Spencer, Mom, and a host of others who provided moral support while I slogged through the thousands of pages of data on my desk.

Introduction

This book puts the basic hardware and software information about Macintosh computers in one place.

What's the difference between the ASCII character definitions and the Macintosh's interpretation of them? Normally you'd have to consult two references to answer that question. Here, the answer is in one place (Tables 1.16 and 1.17). Suppose that you're using TMON or MacNosy to trace instructions in a questionable segment of your program and you come across a trap call that isn't formatted correctly. Where do you go to find the correct call? Did the call change when new models came out, or has Apple issued a bulletin that further describes the idiosyncrasies of the call? And what if it's the underlying data structures you're passing to the call that are wrong? Can you find the answer quickly by consulting one volume? You can now. Again, the answers to all your questions are in this book.

Whatever you need to know about the Macintosh—cable connections, trap calls, data structure formats, internal registers, file formats, resource formats, and so on—you'll find at least the primary information here.

To provide you with a coherent summary of all the technical information scattered throughout the books on your shelves, this book:

- acts as a primary source for most information contained in Apple references
- provides pointers to further information and to items too detailed for complete inclusion here
- organizes information about the Macintosh into four logical groups, each of which has been further organized to help you find answers fast

Let's look at each of these functions individually.

Primary Source of Macintosh Information

Apple's published information about the Macintosh is spread over two dozen volumes at present. Even the basic information (*Inside Macintosh*) takes up six volumes, each with several hundred pages. Worse still, the *Inside Macintosh* material is chronologically organized. Volumes I and II contain the basic information about the original Macintosh models (128K, 512K, XL). Volume III summarizes that information. Volume IV adds material that changed with the introduction of the Mac Plus and the HFS file system. And Volume V adds the Mac SE and Mac II information. To find information about a single trap, you may end up consulting three or four volumes in order to make sure that nothing changed as subsequent models were introduced.

This book distills the most important information from the Apple documentation and other related sources. A single table here may have a half dozen or more sources and is complete and accurate as of the end of 1988. That is why I call this book a "primary" source of information—it's the first source you should turn to when you need to look something up. The full list of primary references used in compiling this work is included in the bibliography at the end of this book.

Pointers to Further Information

Every table in this book points to the original source data. Furthermore, the book is fully cross-referenced so that every table also points to related tables elsewhere in the book.

Pointers to related information are in the *Source* and *See Also* notes at the bottom of each table.

- The *Source* note gives the name and page number(s) of the primary source used in compiling the table.

- The *See Also* note gives the numbers and names of other tables in this book that contain related information you might want to consult. (Read the section titled "Organization" for more on table numbering.)

Every effort has been made to ensure that the page numbers given in the *Source* notes are accurate. Although Apple and Addison-Wesley have published two distinct editions of the *Inside Macintosh* series, the page numbers have been retained in the new editions, so the pointers here should be accurate. Some of the other works referenced—such as the *Apple Numerics Manual*—have undergone sweeping revisions that have changed the page numbering. That's why you should take a close look at Table 0.0, which lists the actual editions used in compiling this book. If another edition has been used for a particular table, that will be specifically stated in the *Source* note. If you have an edition of a work that is not the same as that listed in Table 0.0, you may find some page-number discrepancies in the *Source* notes.

Organization

To help you find information easily, as well as to help you see relationships among tables, this book is organized into four main sections:

- Section 1: General Information
- Section 2: Operating System Software, Resources, and File Formats
- Section 3: ROM Traps and System Calls
- Section 4: Hardware

All tables are numbered. The number to the left of the decimal point indicates the section number; the number to the right of the decimal point indicates the table's place in that section. Thus, table 3.120 refers to the 120nd table in section 3.

In addition, the Table of Contents is a fully indented outline of the entire table list. In Section 3, for example, Memory Manager is indented under Operating System Managers; Memory Manager Routines is indented under Memory Manager. Section 3 also follows the same order of presentation as does Inside Macintosh. If you're familiar with the source material, therefore, you'll have an even easier time finding information here.

How Tables Are Presented

In the Table 0.0 you'll see a representative sample of a table.

At the top are the table number and table name in boldface type. They help you identify the table contents, with the number also serving as the cross-reference used in this book.

If a table has been broken into subtables, a subtable heading appears in italic type immediately above each subtable.

Headings down the left side and across the top of the table are in italic to set them off as headings (as opposed to table data). Where headings are grouped (bit numbers, for example, which are usually in groups of 8 or 16), a group header appears in italic immediately above the first item in the group.

Entries are in (boxed) cells. In cases where groups of data are related, the group appears in a single box with each item on a separate line within the box. Within data entries, several abbreviations are consistently used:

MSB	= most significant bit (or byte)
LSB	= least significant bit (or byte)
LO	= low order
HO	= high order
000	= a binary value of zero, zero, zero
010	= a binary value of zero, one, zero
$1A	= a hexadecimal value of 1A (all hex values are preceded by the $ sign—other numbers are assumed to be decimal)
string	= any group of text characters enclosed in single quotes

0.0. References Used In Compiling This Book

Name	*Edition*	*ISBN # / Part #*	*Publisher*
Apple Numerics Manual	First	0-201-17741-2	Addison-Wesley
Designing Cards and Drivers for the Macintosh II and SE	First	0-201-19256-X	Addison-Wesley
HyperCard Script Language	First	0-201-17632-7	Addison-Wesley
ImageWriter II Technical Reference Manual	First	0-201-17766-8	Addison-Wesley
ImageWriter LQ Reference	First	0-201-17751-X	Addison-Wesley
ImageWriter User's Manual	NA	A9L030	Apple Computer
Inside Macintosh Volume I	First	0-201-17731-5	Addison-Wesley
Inside Macintosh Volume II	First	0-201-17732-3	Addison-Wesley
Inside Macintosh Volume III	First	0-201-17733-1	Addison-Wesley
Inside Macintosh Volume IV	First	0-201-05409-4	Addison-Wesley
Inside Macintosh Volume V	First	0-201-17719-6	Addison-Wesley
Inside Macintosh X-REF	First	0-201-13694-5	Addison-Wesley
LaserWriter Reference	First	0-201-19258-6	Addison-Wesley
Macintosh Family Hardware Reference	First	0-201-19255-1	Addison-Wesley
Macintosh Technical Notes	NA	M7030, M7032, M0058LL/A	Apple Computer
Programmer's Introduction to the Macintosh Family	First	0-201-19254-3	Addison-Wesley
Technical Introduction to the Macintosh Family	First	0-201-17765-X	Addison-Wesley

Notes: New editions exist for the Inside Macintosh series, but the information and page numbers are identical; only the covers have changed.

See Also: Bibliography

char = character (a single byte of information)
dbl word = double word (four bytes)
word = two bytes

Other abbreviations are used in contexts that should be self-explanatory; for example, "res" is short for "resource" in some comments in the resource section.

I have also attempted to retain Apple's naming and identification system. In general, variable and data names start with a lowercase character and the name is an abbreviated representation of the contents (as in ioFilNam, which stands for I/O File Name). Apple isn't always consistent about this, however, and there are some minor deviations here in an attempt to make all data consistent.

At the end of each table is a Notes section. Notes give general information or exceptions related to data in the table as well as footnotes for specific entries. General notes are listed first, followed by footnotes, which are identified with asterisks or other symbols.

SECTION I

GENERAL INFORMATION

1.01. Hexadecimal to Decimal Number Conversion

Byte Values	*Least Significant Digit*															
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
00	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
20	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
30	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
40	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Most Significant Digit *50*	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
60	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
70	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
80	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
90	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A0	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B0	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C0	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D0	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E0	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F0	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

To Use This Table: To determine the decimal value of A4 hex, you look in the row labeled A0 (the most significant hex digit is A) and the column labeled 4 (the least significant hex digit is 4). The value at the intersection of these two headings (164) is the decimal value for A4.

Nibble	
Hex	Dec
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Byte	
Hex	Dec
00	0
10	16
20	32
30	48
40	64
50	80
60	96
70	112
80	128
90	144
A0	160
B0	176
C0	192
D0	208
E0	224
F0	240

Byte+Nibble	
Hex	Dec
000	0
100	256
200	512
300	768
400	1024
500	1280
600	1536
700	1792
800	2048
900	2304
A00	2560
B00	2816
C00	3072
D00	3328
E00	3584
F00	3840

Word	
Hex	Dec
0000	0
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440

To Use This Table: To convert a two-byte (word) hexadecimal value to decimal, find the value associated with each hex digit place in the above table; add these numbers together. For example, the hex value A5D7 would be equal to 40960 (the A000 value) plus 1280 (the 500 value) plus 208 (the D0 value) plus 7, or 42455.

See Also: 1.05. Binary Number Conversions
1.08. Decimal to Hexadecimal Number Conversion

1.02. Hexadecimal to Binary Number Conversion

Byte Values

Hex	Binary	Hex	Binary	Hex	Binary	Hex	Binary
0	0000 0000	40	0100 0000	80	1000 0000	C0	1100 0000
1	0000 0001	41	0100 0001	81	1000 0001	C1	1100 0001
2	0000 0010	42	0100 0010	82	1000 0010	C2	1100 0010
3	0000 0011	43	0100 0011	83	1000 0011	C3	1100 0011
4	0000 0100	44	0100 0100	84	1000 0100	C4	1100 0100
5	0000 0101	45	0100 0101	85	1000 0101	C5	1100 0101
6	0000 0110	46	0100 0110	86	1000 0110	C6	1100 0110
7	0000 0111	47	0100 0111	87	1000 0111	C7	1100 0111
8	0000 1000	48	0100 1000	88	1000 1000	C8	1100 1000
9	0000 1001	49	0100 1001	89	1000 1001	C9	1100 1001
A	0000 1010	4A	0100 1010	8A	1000 1010	CA	1100 1010
B	0000 1011	4B	0100 1011	8B	1000 1011	CB	1100 1011
C	0000 1100	4C	0100 1100	8C	1000 1100	CC	1100 1100
D	0000 1101	4D	0100 1101	8D	1000 1101	CD	1100 1101
E	0000 1110	4E	0100 1110	8E	1000 1110	CE	1100 1110
F	0000 1111	4F	0100 1111	8F	1000 1111	CF	1100 1111
10	0001 0000	50	0101 0000	90	1001 0000	D0	1101 0000
11	0001 0001	51	0101 0001	91	1001 0001	D1	1101 0001
12	0001 0010	52	0101 0010	92	1001 0010	D2	1101 0010
13	0001 0011	53	0101 0011	93	1001 0011	D3	1101 0011
14	0001 0100	54	0101 0100	94	1001 0100	D4	1101 0100
15	0001 0101	55	0101 0101	95	1001 0101	D5	1101 0101
16	0001 0110	56	0101 0110	96	1001 0110	D6	1101 0110
17	0001 0111	57	0101 0111	97	1001 0111	D7	1101 0111
18	0001 1000	58	0101 1000	98	1001 1000	D8	1101 1000
19	0001 1001	59	0101 1001	99	1001 1001	D9	1101 1001
1A	0001 1010	5A	0101 1010	9A	1001 1010	DA	1101 1010
1B	0001 1011	5B	0101 1011	9B	1001 1011	DB	1101 1011
1C	0001 1100	5C	0101 1100	9C	1001 1100	DC	1101 1100
1D	0001 1101	5D	0101 1101	9D	1001 1101	DD	1101 1101
1E	0001 1110	5E	0101 1110	9E	1001 1110	DE	1101 1110
1F	0001 1111	5F	0101 1111	9F	1001 1111	DF	1101 1111
20	0010 0000	60	0110 0000	A0	1010 0000	E0	1110 0000
21	0010 0001	61	0110 0001	A1	1010 0001	E1	1110 0001
22	0010 0010	62	0110 0010	A2	1010 0010	E2	1110 0010
23	0010 0011	63	0110 0011	A3	1010 0011	E3	1110 0011
24	0010 0100	64	0110 0100	A4	1010 0100	E4	1110 0100
25	0010 0101	65	0110 0101	A5	1010 0101	E5	1110 0101
26	0010 0110	66	0110 0110	A6	1010 0110	E6	1110 0110
27	0010 0111	67	0110 0111	A7	1010 0111	E7	1110 0111
28	0010 1000	68	0110 1000	A8	1010 1000	E8	1110 1000
29	0010 1001	69	0110 1001	A9	1010 1001	E9	1110 1001
2A	0010 1010	6A	0110 1010	AA	1010 1010	EA	1110 1010
2B	0010 1011	6B	0110 1011	AB	1010 1011	EB	1110 1011
2C	0010 1100	6C	0110 1100	AC	1010 1100	EC	1110 1100
2D	0010 1101	6D	0110 1101	AD	1010 1101	ED	1110 1101
2E	0010 1110	6E	0110 1110	AE	1010 1110	EE	1110 1110
2F	0010 1111	6F	0110 1111	AF	1010 1111	EF	1110 1111
30	0011 0000	70	0111 0000	B0	1011 0000	F0	1111 0000
31	0011 0001	71	0111 0001	B1	1011 0001	F1	1111 0001
32	0011 0010	72	0111 0010	B2	1011 0010	F2	1111 0010
33	0011 0011	73	0111 0011	B3	1011 0011	F3	1111 0011
34	0011 0100	74	0111 0100	B4	1011 0100	F4	1111 0100
35	0011 0101	75	0111 0101	B5	1011 0101	F5	1111 0101
36	0011 0110	76	0111 0110	B6	1011 0110	F6	1111 0110
37	0011 0111	77	0111 0111	B7	1011 0111	F7	1111 0111
38	0011 1000	78	0111 1000	B8	1011 1000	F8	1111 1000
39	0011 1001	79	0111 1001	B9	1011 1001	F9	1111 1001
3A	0011 1010	7A	0111 1010	BA	1011 1010	FA	1111 1010
3B	0011 1011	7B	0111 1011	BB	1011 1011	FB	1111 1011
3C	0011 1100	7C	0111 1100	BC	1011 1100	FC	1111 1100
3D	0011 1101	7D	0111 1101	BD	1011 1101	FD	1111 1101
3E	0011 1110	7E	0111 1110	BE	1011 1110	FE	1111 1110
3F	0011 1111	7F	0111 1111	BF	1011 1111	FF	1111 1111

To Use This Table: Find the hexadecimal byte value you are looking up in one of the left-most columns and read its binary value in the same row in the column immediately to the right. For example, a hexadecimal value of 2B would result in a binary value of 0010 1011.

Nibble Values

Hex	Binary	Hex	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

To Use This Table: To convert a long hexadecimal number into binary, simply use the table above to substitute for each hexadecimal digit. Thus a hexadecimal value of 9AF2 is 1001 1010 1111 0010 in binary.

See Also: 1.05. Binary Number Conversions
1.07. Decimal to Binary Number Conversion

1.03. Hexadecimal Addition Tables

Results in Hexadecimal

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1		2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2			4	5	6	7	8	9	A	B	C	D	E	F	10	11
3				6	7	8	9	A	B	C	D	E	F	10	11	12
4					8	9	A	B	C	D	E	F	10	11	12	13
5						A	B	C	D	E	F	10	11	12	13	14
6							C	D	E	F	10	11	12	13	14	15
7								E	F	10	11	12	13	14	15	16
8									10	11	12	13	14	15	16	17
9										12	13	14	15	16	17	18
A											14	15	16	17	18	19
B												16	17	18	19	1A
C													18	19	1A	1B
D														1A	1B	1C
E															1C	1D
F																1E

Results in Decimal

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2			4	5	6	7	8	9	10	11	12	13	14	15	16	17
3				6	7	8	9	10	11	12	13	14	15	16	17	18
4					8	9	10	11	12	13	14	15	16	17	18	19
5						10	11	12	13	14	15	16	17	18	19	20
6							12	13	14	15	16	17	18	19	20	21
7								14	15	16	17	18	19	20	21	22
8									16	17	18	19	20	21	22	23
9										18	19	20	21	22	23	24
A											20	21	22	23	24	25
B												22	23	24	25	26
C													24	25	26	27
D														26	27	28
E															28	29
F																30

To Use This Table: Find the column/row intersection of the two hexadecimal nibble values you want to add; the value in that cell is the result of adding the two hex nibbles together.

(continued)

Table 1.03. Continued

Results in Hexadecimal

	10	*20*	*30*	*40*	*50*	*60*	*70*	*80*	*90*	*A0*	*B0*	*C0*	*D0*	*E0*	*F0*
10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0	100
20		40	50	60	70	80	90	A0	B0	C0	D0	E0	F0	100	110
30			60	70	80	90	A0	B0	C0	D0	E0	F0	100	110	120
40				80	90	A0	B0	C0	D0	E0	F0	100	110	120	130
50					A0	B0	C0	D0	E0	F0	100	110	120	130	140
60						C0	D0	E0	F0	100	110	120	130	140	150
70							E0	F0	100	110	120	130	140	150	160
80								100	110	120	130	140	150	160	170
90									120	130	140	150	160	170	180
A0										140	150	160	170	180	190
B0											160	170	180	190	1A0
C0												180	190	1A0	1B0
D0													1A0	1B0	1C0
E0														1C0	1D0
F0															1E0

Results in Decimal

	10	*20*	*30*	*40*	*50*	*60*	*70*	*80*	*90*	*A0*	*B0*	*C0*	*D0*	*E0*	*F0*
10	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256
20		64	80	96	112	128	144	160	176	192	208	224	240	256	272
30			96	112	128	144	160	176	192	208	224	240	256	272	288
40				128	144	160	176	192	208	224	240	256	272	288	304
50					160	176	192	208	224	240	256	272	288	304	320
60						192	208	224	240	256	272	288	304	320	336
70							224	240	256	272	288	304	320	336	352
80								256	272	288	304	320	336	352	368
90									288	304	320	336	352	368	384
A0										320	336	352	368	384	400
B0											352	368	384	400	416
C0												384	400	416	432
D0													416	432	448
E0														448	464
F0															480

To Use This Table: Find the result at the intersection of the column and row that contain the hex numbers you are adding. If you are adding hexadecimal byte values that don't end in 0 (e.g., B4 + A6), first look up the results for the least significant digits (4 + 6 = A), then add this value to the results for the most significant digits (BO + AO = 150, so B4 + A6 = 15A). Remember to carry if necessary (B + B = 16, so BB + AB = 166).

See Also: 1.04. Hexadecimal Multiplication Tables

1.04. Hexadecimal Multiplication Tables

Results in Hexadecimal

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2			4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3				9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4					10	14	18	1C	20	24	28	2C	30	34	38	3C
5						19	1E	23	28	2D	32	37	3C	41	46	4B
6							24	2A	30	36	3C	42	48	4E	54	5A
7								31	38	3F	46	4D	54	5B	62	69
8									40	48	50	58	60	68	70	78
9										51	5A	63	6C	75	7E	87
A											64	6E	78	82	8C	96
B												79	84	8F	9A	A5
C													90	9C	A8	B4
D														A9	B6	C3
E															C4	D2
F																E1

Results in Decimal

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2			4	6	8	10	12	14	16	18	20	22	24	26	28	30
3				9	12	15	18	21	24	27	30	33	36	39	42	45
4					16	20	24	28	32	36	40	44	48	52	56	60
5						25	30	35	40	45	50	55	60	65	70	75
6							36	42	48	54	60	66	72	78	84	90
7								49	56	63	70	77	84	91	98	105
8									64	72	80	88	96	104	112	120
9										81	90	99	108	117	126	135
A											100	110	120	130	140	150
B												121	132	143	154	165
C													144	156	168	180
D														169	182	195
E															196	210
F																225

To Use This Table: Find the column/row intersection of the two hexadecimal nibble values you want to multiply; the value in that cell is the result of multiplying the two hex nibbles together.

See Also: 1.03. Hexadecimal Addition Tables

1.05. Binary Number Conversions

Binary	Decimal	Hex
0000 0000	0	0
0000 0001	1	1
0000 0010	2	2
0000 0011	3	3
0000 0100	4	4
0000 0101	5	5
0000 0110	6	6
0000 0111	7	7
0000 1000	8	8
0000 1001	9	9
0000 1010	10	A
0000 1011	11	B
0000 1100	12	C
0000 1101	13	D
0000 1110	14	E
0000 1111	15	F
0001 0000	16	10
0001 0001	17	11
0001 0010	18	12
0001 0011	19	13
0001 0100	20	14
0001 0101	21	15
0001 0110	22	16
0001 0111	23	17
0001 1000	24	18
0001 1001	25	19
0001 1010	26	1A
0001 1011	27	1B
0001 1100	28	1C
0001 1101	29	1D
0001 1110	30	1E
0001 1111	31	1F
0010 0000	32	20
0010 0001	33	21
0010 0010	34	22
0010 0011	35	23
0010 0100	36	24
0010 0101	37	25
0010 0110	38	26
0010 0111	39	27
0010 1000	40	28
0010 1001	41	29
0010 1010	42	2A
0010 1011	43	2B
0010 1100	44	2C
0010 1101	45	2D
0010 1110	46	2E
0010 1111	47	2F
0011 0000	48	30
0011 0001	49	31
0011 0010	50	32
0011 0011	51	33
0011 0100	52	34
0011 0101	53	35
0011 0110	54	36
0011 0111	55	37
0011 1000	56	38
0011 1001	57	39
0011 1010	58	3A
0011 1011	59	3B
0011 1100	60	3C
0011 1101	61	3D
0011 1110	62	3E
0011 1111	63	3F

Binary	Decimal	Hex
0100 0000	64	40
0100 0001	65	41
0100 0010	66	42
0100 0011	67	43
0100 0100	68	44
0100 0101	69	45
0100 0110	70	46
0100 0111	71	47
0100 1000	72	48
0100 1001	73	49
0100 1010	74	4A
0100 1011	75	4B
0100 1100	76	4C
0100 1101	77	4D
0100 1110	78	4E
0100 1111	79	4F
0101 0000	80	50
0101 0001	81	51
0101 0010	82	52
0101 0011	83	53
0101 0100	84	54
0101 0101	85	55
0101 0110	86	56
0101 0111	87	57
0101 1000	88	58
0101 1001	89	59
0101 1010	90	5A
0101 1011	91	5B
0101 1100	92	5C
0101 1101	93	5D
0101 1110	94	5E
0101 1111	95	5F
0110 0000	96	60
0110 0001	97	61
0110 0010	98	62
0110 0011	99	63
0110 0100	100	64
0110 0101	101	65
0110 0110	102	66
0110 0111	103	67
0110 1000	104	68
0110 1001	105	69
0110 1010	106	6A
0110 1011	107	6B
0110 1100	108	6C
0110 1101	109	6D
0110 1110	110	6E
0110 1111	111	6F
0111 0000	112	70
0111 0001	113	71
0111 0010	114	72
0111 0011	115	73
0111 0100	116	74
0111 0101	117	75
0111 0110	118	76
0111 0111	119	77
0111 1000	120	78
0111 1001	121	79
0111 1010	122	7A
0111 1011	123	7B
0111 1100	124	7C
0111 1101	125	7D
0111 1110	126	7E
0111 1111	127	7F

Binary	Decimal	Hex
1000 0000	128	80
1000 0001	129	81
1000 0010	130	82
1000 0011	131	83
1000 0100	132	84
1000 0101	133	85
1000 0110	134	86
1000 0111	135	87
1000 1000	136	88
1000 1001	137	89
1000 1010	138	8A
1000 1011	139	8B
1000 1100	140	8C
1000 1101	141	8D
1000 1110	142	8E
1000 1111	143	8F
1001 0000	144	90
1001 0001	145	91
1001 0010	146	92
1001 0011	147	93
1001 0100	148	94
1001 0101	149	95
1001 0110	150	96
1001 0111	151	97
1001 1000	152	98
1001 1001	153	99
1001 1010	154	9A
1001 1011	155	9B
1001 1100	156	9C
1001 1101	157	9D
1001 1110	158	9E
1001 1111	159	9F
1010 0000	160	A0
1010 0001	161	A1
1010 0010	162	A2
1010 0011	163	A3
1010 0100	164	A4
1010 0101	165	A5
1010 0110	166	A6
1010 0111	167	A7
1010 1000	168	A8
1010 1001	169	A9
1010 1010	170	AA
1010 1011	171	AB
1010 1100	172	AC
1010 1101	173	AD
1010 1110	174	AE
1010 1111	175	AF
1011 0000	176	B0
1011 0001	177	B1
1011 0010	178	B2
1011 0011	179	B3
1011 0100	180	B4
1011 0101	181	B5
1011 0110	182	B6
1011 0111	183	B7
1011 1000	184	B8
1011 1001	185	B9
1011 1010	186	BA
1011 1011	187	BB
1011 1100	188	BC
1011 1101	189	BD
1011 1110	190	BE
1011 1111	191	BF

Binary	Decimal	Hex
1100 0000	192	C0
1100 0001	193	C1
1100 0010	194	C2
1100 0011	195	C3
1100 0100	196	C4
1100 0101	197	C5
1100 0110	198	C6
1100 0111	199	C7
1100 1000	200	C8
1100 1001	201	C9
1100 1010	202	CA
1100 1011	203	CB
1100 1100	204	CC
1100 1101	205	CD
1100 1110	206	CE
1100 1111	207	CF
1101 0000	208	D0
1101 0001	209	D1
1101 0010	210	D2
1101 0011	211	D3
1101 0100	212	D4
1101 0101	213	D5
1101 0110	214	D6
1101 0111	215	D7
1101 1000	216	D8
1101 1001	217	D9
1101 1010	218	DA
1101 1011	219	DB
1101 1100	220	DC
1101 1101	221	DD
1101 1110	222	DE
1101 1111	223	DF
1110 0000	224	E0
1110 0001	225	E1
1110 0010	226	E2
1110 0011	227	E3
1110 0100	228	E4
1110 0101	229	E5
1110 0110	230	E6
1110 0111	231	E7
1110 1000	232	E8
1110 1001	233	E9
1110 1010	234	EA
1110 1011	235	EB
1110 1100	236	EC
1110 1101	237	ED
1110 1110	238	EE
1110 1111	239	EF
1111 0000	240	F0
1111 0001	241	F1
1111 0010	242	F2
1111 0011	243	F3
1111 0100	244	F4
1111 0101	245	F5
1111 0110	246	F6
1111 0111	247	F7
1111 1000	248	F8
1111 1001	249	F9
1111 1010	250	FA
1111 1011	251	FB
1111 1100	252	FC
1111 1101	253	FD
1111 1110	254	FE
1111 1111	255	FF

To Use This Table: Find the value you want to look up by locating its binary equivalent in one of the left-most columns. Read the converted value in the appropriate column on the same row; e.g., the hexadecimal equivalent of binary 1110 is E (the third column).

See Also:

1.02. Hexadecimal to Binary Number Conversion
1.07. Decimal to Binary Number Conversion

1.06. Binary to Signed Decimal Number Conversion

Binary	Decimal	Binary	Decimal	Binary	Decimal	Binary	Decimal
0000 0000	0	0100 0000	64	1000 0000	-128	1100 0000	-64
0000 0001	1	0100 0001	65	1000 0001	-127	1100 0001	-63
0000 0010	2	0100 0010	66	1000 0010	-126	1100 0010	-62
0000 0011	3	0100 0011	67	1000 0011	-125	1100 0011	-61
0000 0100	4	0100 0100	68	1000 0100	-124	1100 0100	-60
0000 0101	5	0100 0101	69	1000 0101	-123	1100 0101	-59
0000 0110	6	0100 0110	70	1000 0110	-122	1100 0110	-58
0000 0111	7	0100 0111	71	1000 0111	-121	1100 0111	-57
0000 1000	8	0100 1000	72	1000 1000	-120	1100 1000	-56
0000 1001	9	0100 1001	73	1000 1001	-119	1100 1001	-55
0000 1010	10	0100 1010	74	1000 1010	-118	1100 1010	-54
0000 1011	11	0100 1011	75	1000 1011	-117	1100 1011	-53
0000 1100	12	0100 1100	76	1000 1100	-116	1100 1100	-52
0000 1101	13	0100 1101	77	1000 1101	-115	1100 1101	-51
0000 1110	14	0100 1110	78	1000 1110	-114	1100 1110	-50
0000 1111	15	0100 1111	79	1000 1111	-113	1100 1111	-49
0001 0000	16	0101 0000	80	1001 0000	-112	1101 0000	-48
0001 0001	17	0101 0001	81	1001 0001	-111	1101 0001	-47
0001 0010	18	0101 0010	82	1001 0010	-110	1101 0010	-46
0001 0011	19	0101 0011	83	1001 0011	-109	1101 0011	-45
0001 0100	20	0101 0100	84	1001 0100	-108	1101 0100	-44
0001 0101	21	0101 0101	85	1001 0101	-107	1101 0101	-43
0001 0110	22	0101 0110	86	1001 0110	-106	1101 0110	-42
0001 0111	23	0101 0111	87	1001 0111	-105	1101 0111	-41
0001 1000	24	0101 1000	88	1001 1000	-104	1101 1000	-40
0001 1001	25	0101 1001	89	1001 1001	-103	1101 1001	-39
0001 1010	26	0101 1010	90	1001 1010	-102	1101 1010	-38
0001 1011	27	0101 1011	91	1001 1011	-101	1101 1011	-37
0001 1100	28	0101 1100	92	1001 1100	-100	1101 1100	-36
0001 1101	29	0101 1101	93	1001 1101	-99	1101 1101	-35
0001 1110	30	0101 1110	94	1001 1110	-98	1101 1110	-34
0001 1111	31	0101 1111	95	1001 1111	-97	1101 1111	-33
0010 0000	32	0110 0000	96	1010 0000	-96	1110 0000	-32
0010 0001	33	0110 0001	97	1010 0001	-95	1110 0001	-31
0010 0010	34	0110 0010	98	1010 0010	-94	1110 0010	-30
0010 0011	35	0110 0011	99	1010 0011	-93	1110 0011	-29
0010 0100	36	0110 0100	100	1010 0100	-92	1110 0100	-28
0010 0101	37	0110 0101	101	1010 0101	-91	1110 0101	-27
0010 0110	38	0110 0110	102	1010 0110	-90	1110 0110	-26
0010 0111	39	0110 0111	103	1010 0111	-89	1110 0111	-25
0010 1000	40	0110 1000	104	1010 1000	-88	1110 1000	-24
0010 1001	41	0110 1001	105	1010 1001	-87	1110 1001	-23
0010 1010	42	0110 1010	106	1010 1010	-86	1110 1010	-22
0010 1011	43	0110 1011	107	1010 1011	-85	1110 1011	-21
0010 1100	44	0110 1100	108	1010 1100	-84	1110 1100	-20
0010 1101	45	0110 1101	109	1010 1101	-83	1110 1101	-19
0010 1110	46	0110 1110	110	1010 1110	-82	1110 1110	-18
0010 1111	47	0110 1111	111	1010 1111	-81	1110 1111	-17
0011 0000	48	0111 0000	112	1011 0000	-80	1111 0000	-16
0011 0001	49	0111 0001	113	1011 0001	-79	1111 0001	-15
0011 0010	50	0111 0010	114	1011 0010	-78	1111 0010	-14
0011 0011	51	0111 0011	115	1011 0011	-77	1111 0011	-13
0011 0100	52	0111 0100	116	1011 0100	-76	1111 0100	-12
0011 0101	53	0111 0101	117	1011 0101	-75	1111 0101	-11
0011 0110	54	0111 0110	118	1011 0110	-74	1111 0110	-10
0011 0111	55	0111 0111	119	1011 0111	-73	1111 0111	-9
0011 1000	56	0111 1000	120	1011 1000	-72	1111 1000	-8
0011 1001	57	0111 1001	121	1011 1001	-71	1111 1001	-7
0011 1010	58	0111 1010	122	1011 1010	-70	1111 1010	-6
0011 1011	59	0111 1011	123	1011 1011	-69	1111 1011	-5
0011 1100	60	0111 1100	124	1011 1100	-68	1111 1100	-4
0011 1101	61	0111 1101	125	1011 1101	-67	1111 1101	-3
0011 1110	62	0111 1110	126	1011 1110	-66	1111 1110	-2
0011 1111	63	0111 1111	127	1011 1111	-65	1111 1111	-1

To Use This Table: Find the binary value you want to convert in one of the left columns. Read its corresponding signed decimal value in the same row in the column to the right.

See Also: 1.05. Binary Number Conversions

1.07. Decimal to Binary Number Conversion

Dec	Binary	Dec	Binary	Dec	Binary	Dec	Binary
0	0000 0000	64	0100 0000	128	1000 0000	192	1100 0000
1	0000 0001	65	0100 0001	129	1000 0001	193	1100 0001
2	0000 0010	66	0100 0010	130	1000 0010	194	1100 0010
3	0000 0011	67	0100 0011	131	1000 0011	195	1100 0011
4	0000 0100	68	0100 0100	132	1000 0100	196	1100 0100
5	0000 0101	69	0100 0101	133	1000 0101	197	1100 0101
6	0000 0110	70	0100 0110	134	1000 0110	198	1100 0110
7	0000 0111	71	0100 0111	135	1000 0111	199	1100 0111
8	0000 1000	72	0100 1000	136	1000 1000	200	1100 1000
9	0000 1001	73	0100 1001	137	1000 1001	201	1100 1001
10	0000 1010	74	0100 1010	138	1000 1010	202	1100 1010
11	0000 1011	75	0100 1011	139	1000 1011	203	1100 1011
12	0000 1100	76	0100 1100	140	1000 1100	204	1100 1100
13	0000 1101	77	0100 1101	141	1000 1101	205	1100 1101
14	0000 1110	78	0100 1110	142	1000 1110	206	1100 1110
15	0000 1111	79	0100 1111	143	1000 1111	207	1100 1111
16	0001 0000	80	0101 0000	144	1001 0000	208	1101 0000
17	0001 0001	81	0101 0001	145	1001 0001	209	1101 0001
18	0001 0010	82	0101 0010	146	1001 0010	210	1101 0010
19	0001 0011	83	0101 0011	147	1001 0011	211	1101 0011
20	0001 0100	84	0101 0100	148	1001 0100	212	1101 0100
21	0001 0101	85	0101 0101	149	1001 0101	213	1101 0101
22	0001 0110	86	0101 0110	150	1001 0110	214	1101 0110
23	0001 0111	87	0101 0111	151	1001 0111	215	1101 0111
24	0001 1000	88	0101 1000	152	1001 1000	216	1101 1000
25	0001 1001	89	0101 1001	153	1001 1001	217	1101 1001
26	0001 1010	90	0101 1010	154	1001 1010	218	1101 1010
27	0001 1011	91	0101 1011	155	1001 1011	219	1101 1011
28	0001 1100	92	0101 1100	156	1001 1100	220	1101 1100
29	0001 1101	93	0101 1101	157	1001 1101	221	1101 1101
30	0001 1110	94	0101 1110	158	1001 1110	222	1101 1110
31	0001 1111	95	0101 1111	159	1001 1111	223	1101 1111
32	0010 0000	96	0110 0000	160	1010 0000	224	1110 0000
33	0010 0001	97	0110 0001	161	1010 0001	225	1110 0001
34	0010 0010	98	0110 0010	162	1010 0010	226	1110 0010
35	0010 0011	99	0110 0011	163	1010 0011	227	1110 0011
36	0010 0100	100	0110 0100	164	1010 0100	228	1110 0100
37	0010 0101	101	0110 0101	165	1010 0101	229	1110 0101
38	0010 0110	102	0110 0110	166	1010 0110	230	1110 0110
39	0010 0111	103	0110 0111	167	1010 0111	231	1110 0111
40	0010 1000	104	0110 1000	168	1010 1000	232	1110 1000
41	0010 1001	105	0110 1001	169	1010 1001	233	1110 1001
42	0010 1010	106	0110 1010	170	1010 1010	234	1110 1010
43	0010 1011	107	0110 1011	171	1010 1011	235	1110 1011
44	0010 1100	108	0110 1100	172	1010 1100	236	1110 1100
45	0010 1101	109	0110 1101	173	1010 1101	237	1110 1101
46	0010 1110	110	0110 1110	174	1010 1110	238	1110 1110
47	0010 1111	111	0110 1111	175	1010 1111	239	1110 1111
48	0011 0000	112	0111 0000	176	1011 0000	240	1111 0000
49	0011 0001	113	0111 0001	177	1011 0001	241	1111 0001
50	0011 0010	114	0111 0010	178	1011 0010	242	1111 0010
51	0011 0011	115	0111 0011	179	1011 0011	243	1111 0011
52	0011 0100	116	0111 0100	180	1011 0100	244	1111 0100
53	0011 0101	117	0111 0101	181	1011 0101	245	1111 0101
54	0011 0110	118	0111 0110	182	1011 0110	246	1111 0110
55	0011 0111	119	0111 0111	183	1011 0111	247	1111 0111
56	0011 1000	120	0111 1000	184	1011 1000	248	1111 1000
57	0011 1001	121	0111 1001	185	1011 1001	249	1111 1001
58	0011 1010	122	0111 1010	186	1011 1010	250	1111 1010
59	0011 1011	123	0111 1011	187	1011 1011	251	1111 1011
60	0011 1100	124	0111 1100	188	1011 1100	252	1111 1100
61	0011 1101	125	0111 1101	189	1011 1101	253	1111 1101
62	0011 1110	126	0111 1110	190	1011 1110	254	1111 1110
63	0011 1111	127	0111 1111	191	1011 1111	255	1111 1111

To Use This Table: Find the decimal byte value you are looking up in one of left columns and read its binary value in the same row in the column immediately to the right. For example, a decimal value of 43 would result in a binary value of 0010 1011.

See Also: 1.05. Binary Number Conversions

1.08. Decimal to Hexadecimal Number Conversion

Dec	Hex
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A

Dec	Hex
27	1B
28	1C
29	1D
30	1E
31	1F
32	20
33	21
34	22
35	23
36	24
37	25
38	26
39	27
40	28
41	29
42	2A
43	2B
44	2C
45	2D
46	2E
47	2F
48	30
49	31
50	32
51	33
52	34

Dec	Hex
53	35
54	36
55	37
56	38
57	39
58	3A
59	3B
60	3C
61	3D
62	3E
63	3F
64	40
65	41
66	42
67	43
68	44
69	45
70	46
71	47
72	48
73	49
74	4A
75	4B
76	4C
77	4D
78	4E

Dec	Hex
79	4F
80	50
81	51
82	52
83	53
84	54
85	55
86	56
87	57
88	58
89	59
90	5A
91	5B
92	5C
93	5D
94	5E
95	5F
96	60
97	61
98	62
99	63
100	64
200	C8
300	12C
400	190
500	1F4

Dec	Hex
600	258
700	2BC
800	320
900	384
1000	3E8
2000	7D0
3000	BB8
4000	FA0
5000	1388
6000	1770
7000	1B58
8000	1F40
9000	2328
10000	2710
20000	4E20
30000	7530
40000	9C40
50000	C350
60000	EA60
70000	11170
80000	13880
90000	15F90
100000	186A0

To Use This Table: Find the decimal value in the left column and read the corresponding hexadecimal value in the same row in the column to its right. If you are converting a decimal number larger than 100, you may have to perform the conversion in steps, adding the results together. For example, to convert 12345 into hex, first obtain the hex value of decimal 10000 (2710H), then add this to the value for 2000 decimal (7D0H), then add this to the value for 300 decimal (12CH), then add this to the value for 45 decimal (2DH). The proper result is 3039H; remember that the numbers you are adding are in hexadecimal.

See Also: 1.01. Hexadecimal to Decimal Number Conversion
1.03. Hexadecimal Addition Tables

1.09. Two's Complements

Binary	Complement
1111 1111	0000 0001
1111 1110	0000 0010
1111 1101	0000 0011
1111 1100	0000 0100
1111 1011	0000 0101
1111 1010	0000 0110
1111 1001	0000 0111
1111 1000	0000 1000
1111 0111	0000 1001
1111 0110	0000 1010
1111 0101	0000 1011
1111 0100	0000 1100
1111 0011	0000 1101
1111 0010	0000 1110
1111 0001	0000 1111
1111 0000	0001 0000
1110 1111	0001 0001
1110 1110	0001 0010
1110 1101	0001 0011
1110 1100	0001 0100
1110 1011	0001 0101
1110 1010	0001 0110
1110 1001	0001 0111
1110 1000	0001 1000
1110 0111	0001 1001
1110 0110	0001 1010
1110 0101	0001 1011
1110 0100	0001 1100
1110 0011	0001 1101
1110 0010	0001 1110
1110 0001	0001 1111
1110 0000	0010 0000
1101 1111	0010 0001
1101 1110	0010 0010
1101 1101	0010 0011
1101 1100	0010 0100
1101 1011	0010 0101
1101 1010	0010 0110
1101 1001	0010 0111
1101 1000	0010 1000
1101 0111	0010 1001
1101 0110	0010 1010
1101 0101	0010 1011
1101 0100	0010 1100
1101 0011	0010 1101
1101 0010	0010 1110
1101 0001	0010 1111
1101 0000	0011 0000
1100 1111	0011 0001
1100 1110	0011 0010
1100 1101	0011 0011
1100 1100	0011 0100
1100 1011	0011 0101
1100 1010	0011 0110
1100 1001	0011 0111
1100 1000	0011 1000
1100 0111	0011 1001
1100 0110	0011 1010
1100 0101	0011 1011
1100 0100	0011 1100
1100 0011	0011 1101
1100 0010	0011 1110
1100 0001	0011 1111
1100 0000	0100 0000

Binary	Complement
1011 1111	0100 0001
1011 1110	0100 0010
1011 1101	0100 0011
1011 1100	0100 0100
1011 1011	0100 0101
1011 1010	0100 0110
1011 1001	0100 0111
1011 1000	0100 1000
1011 0111	0100 1001
1011 0110	0100 1010
1011 0101	0100 1011
1011 0100	0100 1100
1011 0011	0100 1101
1011 0010	0100 1110
1011 0001	0100 1111
1011 0000	0101 0000
1010 1111	0101 0001
1010 1110	0101 0010
1010 1101	0101 0011
1010 1100	0101 0100
1010 1011	0101 0101
1010 1010	0101 0110
1010 1001	0101 0111
1010 1000	0101 1000
1010 0111	0101 1001
1010 0110	0101 1010
1010 0101	0101 1011
1010 0100	0101 1100
1010 0011	0101 1101
1010 0010	0101 1110
1010 0001	0101 1111
1010 0000	0110 0000
1001 1111	0110 0001
1001 1110	0110 0010
1001 1101	0110 0011
1001 1100	0110 0100
1001 1011	0110 0101
1001 1010	0110 0110
1001 1001	0110 0111
1001 1000	0110 1000
1001 0111	0110 1001
1001 0110	0110 1010
1001 0101	0110 1011
1001 0100	0110 1100
1001 0011	0110 1101
1001 0010	0110 1110
1001 0001	0110 1111
1001 0000	0111 0000
1000 1111	0111 0001
1000 1110	0111 0010
1000 1101	0111 0011
1000 1100	0111 0100
1000 1011	0111 0101
1000 1010	0111 0110
1000 1001	0111 0111
1000 1000	0111 1000
1000 0111	0111 1001
1000 0110	0111 1010
1000 0101	0111 1011
1000 0100	0111 1100
1000 0011	0111 1101
1000 0010	0111 1110
1000 0001	0111 1111
1000 0000	1000 0000

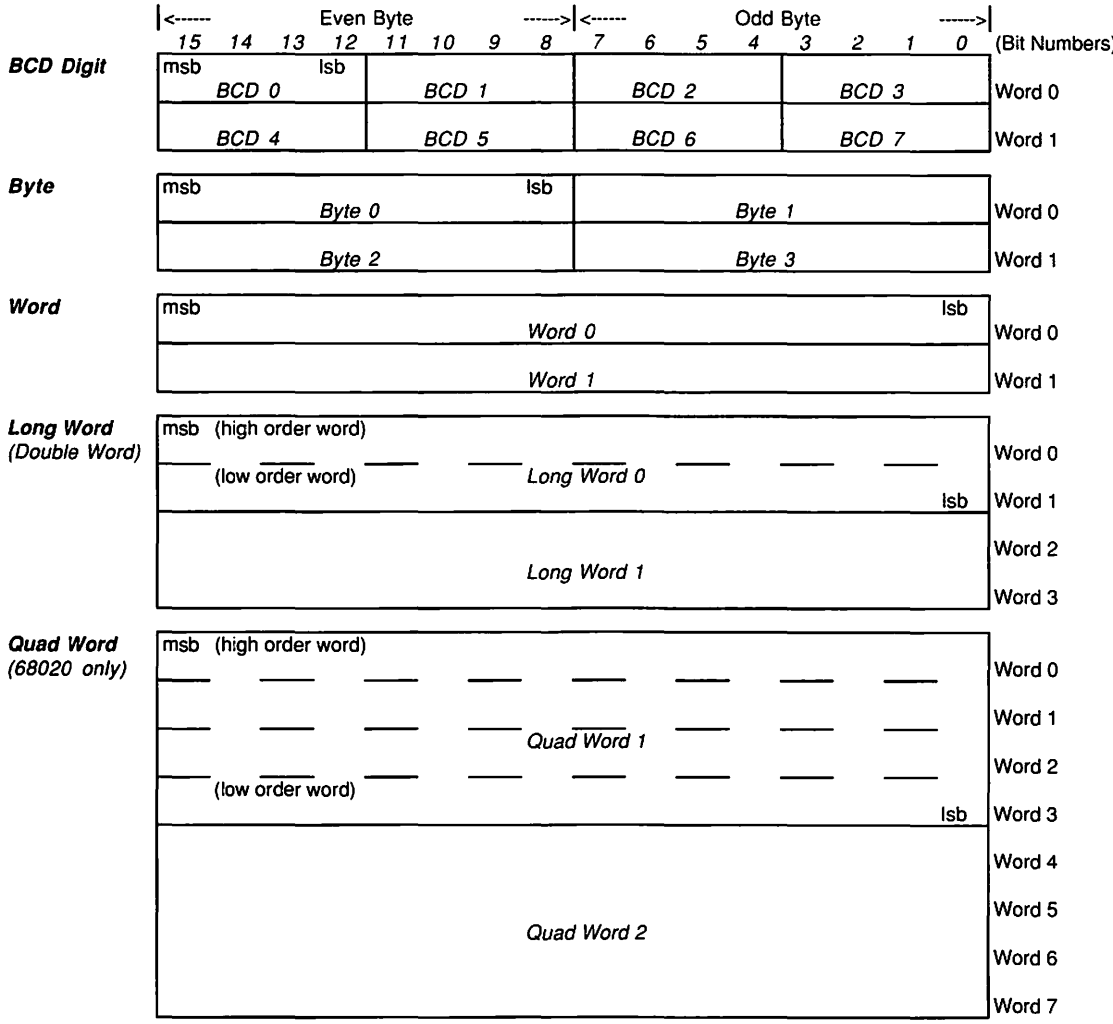
Binary	Complement
0111 1111	1000 0001
0111 1110	1000 0010
0111 1101	1000 0011
0111 1100	1000 0100
0111 1011	1000 0101
0111 1010	1000 0110
0111 1001	1000 0111
0111 1000	1000 1000
0111 0111	1000 1001
0111 0110	1000 1010
0111 0101	1000 1011
0111 0100	1000 1100
0111 0011	1000 1101
0111 0010	1000 1110
0111 0001	1000 1111
0111 0000	1001 0000
0110 1111	1001 0001
0110 1110	1001 0010
0110 1101	1001 0011
0110 1100	1001 0100
0110 1011	1001 0101
0110 1010	1001 0110
0110 1001	1001 0111
0110 1000	1001 1000
0110 0111	1001 1001
0110 0110	1001 1010
0110 0101	1001 1011
0110 0100	1001 1100
0110 0011	1001 1101
0110 0010	1001 1110
0110 0001	1001 1111
0110 0000	1010 0000
0101 1111	1010 0001
0101 1110	1010 0010
0101 1101	1010 0011
0101 1100	1010 0100
0101 1011	1010 0101
0101 1010	1010 0110
0101 1001	1010 0111
0101 1000	1010 1000
0101 0111	1010 1001
0101 0110	1010 1010
0101 0101	1010 1011
0101 0100	1010 1100
0101 0011	1010 1101
0101 0010	1010 1110
0101 0001	1010 1111
0101 0000	1011 0000
0100 1111	1011 0001
0100 1110	1011 0010
0100 1101	1011 0011
0100 1100	1011 0100
0100 1011	1011 0101
0100 1010	1011 0110
0100 1001	1011 0111
0100 1000	1011 1000
0100 0111	1011 1001
0100 0110	1011 1010
0100 0101	1011 1011
0100 0100	1011 1100
0100 0011	1011 1101
0100 0010	1011 1110
0100 0001	1011 1111
0100 0000	1100 0000

Binary	Complement
0011 1111	1100 0001
0011 1110	1100 0010
0011 1101	1100 0011
0011 1100	1100 0100
0011 1011	1100 0101
0011 1010	1100 0110
0011 1001	1100 0111
0011 1000	1100 1000
0011 0111	1100 1001
0011 0110	1100 1010
0011 0101	1100 1011
0011 0100	1100 1100
0011 0011	1100 1101
0011 0010	1100 1110
0011 0001	1100 1111
0011 0000	1101 0000
0010 1111	1101 0001
0010 1110	1101 0010
0010 1101	1101 0011
0010 1100	1101 0100
0010 1011	1101 0101
0010 1010	1101 0110
0010 1001	1101 0111
0010 1000	1101 1000
0010 0111	1101 1001
0010 0110	1101 1010
0010 0101	1101 1011
0010 0100	1101 1100
0010 0011	1101 1101
0010 0010	1101 1110
0010 0001	1101 1111
0010 0000	1110 0000
0001 1111	1110 0001
0001 1110	1110 0010
0001 1101	1110 0011
0001 1100	1110 0100
0001 1011	1110 0101
0001 1010	1110 0110
0001 1001	1110 0111
0001 1000	1110 1000
0001 0111	1110 1001
0001 0110	1110 1010
0001 0101	1110 1011
0001 0100	1110 1100
0001 0011	1110 1101
0001 0010	1110 1110
0001 0001	1110 1111
0001 0000	1111 0000
0000 1111	1111 0001
0000 1110	1111 0010
0000 1101	1111 0011
0000 1100	1111 0100
0000 1011	1111 0101
0000 1010	1111 0110
0000 1001	1111 0111
0000 1000	1111 1000
0000 0111	1111 1001
0000 0110	1111 1010
0000 0101	1111 1011
0000 0100	1111 1100
0000 0011	1111 1101
0000 0010	1111 1110
0000 0001	1111 1111
0000 0000	0000 0000

To Use This Table: Find the binary value you want to convert in one of the left columns. Read its corresponding two's complement value in the same row in the column to the right.

See Also: 1.05. Binary Number Conversions

1.10. Common 680x0 Data Formats



Integer Storage Abilities

	Smallest Integer Value	Largest Integer Value
BCD Digit	0	15
Byte	-128	127
Word	-32,768	32,767
Long Word	-2,147,483,648	2,147,483,647
Quad Word	-9,223,372,036,854,775,808	9,223,372,036,854,775,807

Notes: lsb = least significant bit, msb = most significant bit.

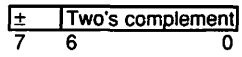
See Also: 1.11. Common Macintosh Data Formats

Source: M68000 Programmer's Reference Manual (5th Edition), pages 2-1 to 2-4
MC68020 32-Bit Microprocessor User's Manual (2nd Edition), pages 1-4, 2-1 to 2-4

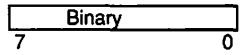
1.11. Common Macintosh Data Formats

GENERAL DATA TYPES

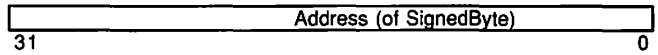
SignedByte



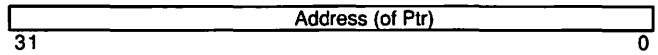
Byte



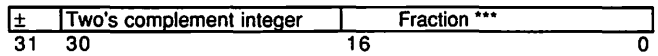
Ptr* Pointer to SignedByte



Handle Pointer to Ptr

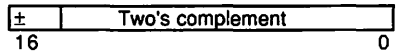


Fixed

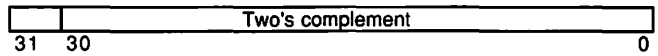


PASCAL DATA TYPES

Int



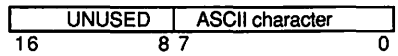
Longint



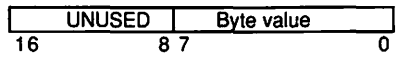
Boolean



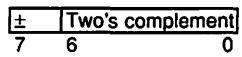
Char



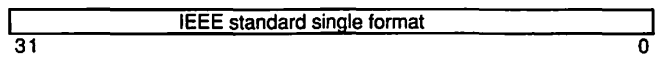
Byte



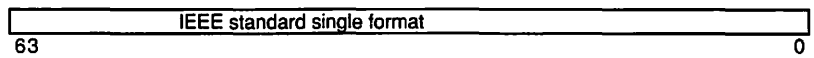
Signed Byte



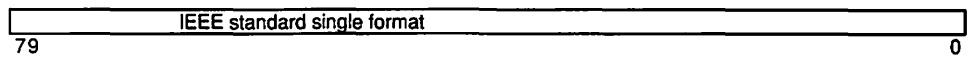
Single



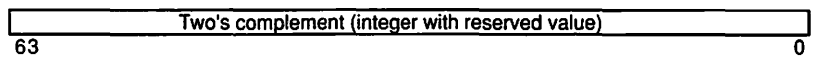
Double



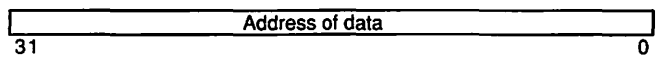
Extended



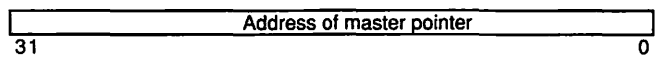
Computational



Ptr



Handle

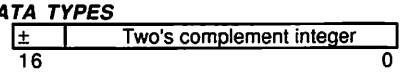


(continued)

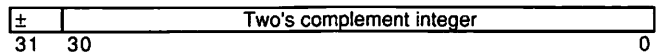
Table 1.11. Continued

STACK-BASED TRAP DATA TYPES

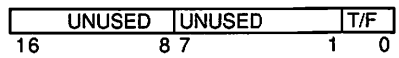
Int



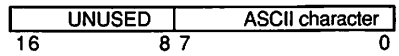
Longint



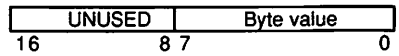
Boolean



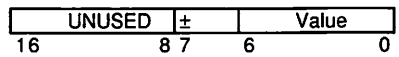
Char



Byte



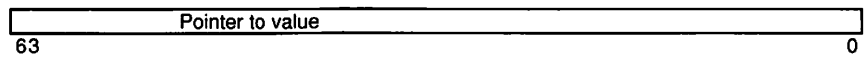
Signed Byte



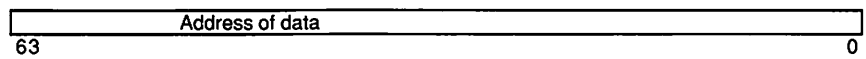
Single, Double, or Computational



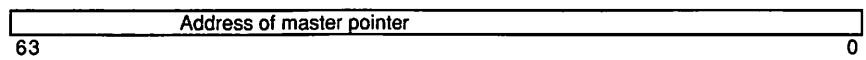
Extended



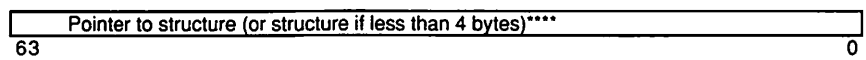
Ptr



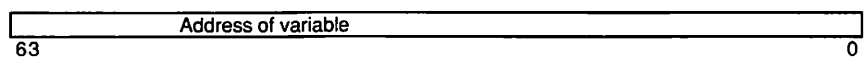
Handle



Record or Array



VAR Parameter



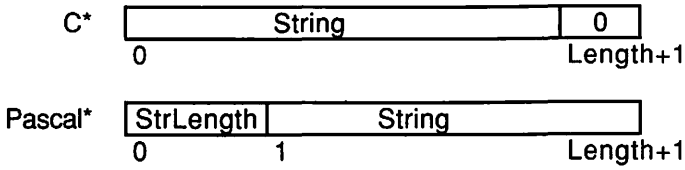
Notes:

- Numbers beneath boxes indicate bit numbers (high = most significant).
- A bit value of 1 in the sign position (±) indicates the value is negative.
- The IEEE floating-point format has an assumed high-order bit of 1 (i.e., is normalized).
- Note that the exponent for IEEE floating-point numbers is "biased" by an implementation-independent amount. For the 680x0, the real exponent = exponent - 1023.
- *If the pointer is to a specific type other than Signed Byte, the name of this data type will be TypePtr, as in StringPtr, FixedPtr, and so on.
- **If the handle is to a specific type other than Signed Byte, the name of this data type will be TypeHandle, as in StringHandle, FixedHandle, and so on.
- ***A binary value is stored in this word, with the bit order reversed from normal binary order (bit 15 is 2's position, bit 14 is 4's position, and so on). Fraction is obtained by dividing the binary value into 1 (e.g. 1 + 16 if bit 12 is set).
- ****If structure is less than 2 bytes in length, the structure will be stored in 32 bits instead of 64; all structures should be padded to the nearest word boundary.

Source: Inside Macintosh, pages I-78 to I-79, I-86 to I-87, I-91

See Also: 1.09. Two's Complements
1.10. Common 680x0 Data Formats

1.12. Common String Formats



Notes: *Not all C and Pascal compilers follow these formats exactly, but this is the recognized standard for each.

Source: Inside Macintosh, page I-86

See Also: 1.10. Common 680x0 Data Formats
1.11. Common Macintosh Data Formats

1.13. Common Memory Area Terminology

Term	*Bits*	*Possible Values*	*Description*	*Conventional Use*
Bit	1	2	Binary digit—a single digital element	Boolean value
Nibble	4	16	One-half of a byte is a "nibble"	BCD digit (0–9) or hex digit (0–F)
Byte	8	256	Standard "cell" of data, especially ASCII characters	ASCII character
Word	16	65,536	68000 CPU deals with this amount of data at a time	Short integer; memory address (not including segment)
Long word	32	4,294,967,296	Smallest memory area that can handle a 680x0 address	Long integers or segment addresses
Quad word	64	NA		Used to contain results of Long Word multiplications
Kilobyte	NA	NA	1024 bytes	NA
Megabyte	NA	NA	1048576 bytes	NA

See Also: 1.19. Powers of Two

1.14. Binary Coded Decimal Number Format

Nibble (one BCD value)

Decimal	*BCD*
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Byte (two BCD values)

Decimal	*BCD*	*Decimal*	*BCD*	*Decimal*	*BCD*	*Decimal*	*BCD*
00	0000 0000	25	0010 0101	50	0101 0000	75	0111 0101
01	0000 0001	26	0010 0110	51	0101 0001	76	0111 0110
02	0000 0010	27	0010 0111	52	0101 0010	77	0111 0111
03	0000 0011	28	0010 1000	53	0101 0011	78	0111 1000
04	0000 0100	29	0010 1001	54	0101 0100	79	0111 1001
05	0000 0101	30	0011 0000	55	0101 0101	80	1000 0000
06	0000 0110	31	0011 0001	56	0101 0110	81	1000 0001
07	0000 0111	32	0011 0010	57	0101 0111	82	1000 0010
08	0000 1000	33	0011 0011	58	0101 1000	83	1000 0011
09	0000 1001	34	0011 0100	59	0101 1001	84	1000 0100
10	0001 0000	35	0011 0101	60	0110 0000	85	1000 0101
11	0001 0001	36	0011 0110	61	0110 0001	86	1000 0110
12	0001 0010	37	0011 0111	62	0110 0010	87	1000 0111
13	0001 0011	38	0011 1000	63	0110 0011	88	1000 1000
14	0001 0100	39	0011 1001	64	0110 0100	89	1000 1001
15	0001 0101	40	0100 0000	65	0110 0101	90	1001 0000
16	0001 0110	41	0100 0001	66	0110 0110	91	1001 0001
17	0001 0111	42	0100 0010	67	0110 0111	92	1001 0010
18	0001 1000	43	0100 0011	68	0110 1000	93	1001 0011
19	0001 1001	44	0100 0100	69	0110 1001	94	1001 0100
20	0010 0000	45	0100 0101	70	0111 0000	95	1001 0101
21	0010 0001	46	0100 0110	71	0111 0001	96	1001 0110
22	0010 0010	47	0100 0111	72	0111 0010	97	1001 0111
23	0010 0011	48	0100 1000	73	0111 0011	98	1001 1000
24	0010 0100	49	0100 1001	74	0111 0100	99	1001 1001

Notes: Two binary coded digits may be stored in one byte, as shown in the lower table.

See Also: 1.07. Decimal to Binary Number Conversion
1.10. Common 680x0 Data Formats
1.11. Common Macintosh Data Formats

1.15. ASCII Control Codes

Dec	Hex	Binary	Mnemonic	Name	Definition
0	00	0000 0000	NUL	Null	Space filler character; used in output timing for some device drivers
1	01	0000 0001	SOH	Start of header	Marks beginning of message header
2	02	0000 0010	STX	Start of text	Marks beginning of data block (text)
3	03	0000 0011	ETX	End of text	Marks end of data block (text)
4	04	0000 0100	EOT	End of transmission	Marks end of transmission session
5	05	0000 0101	ENQ	Inquiry	Request for identification or information
6	06	0000 0110	ACK	Acknowledgement	"Yes" answer to queries or "ready for next transmission"; used in asynchronous protocols for timing
7	07	0000 0111	BEL	Bell	Rings bell or audible alarm on terminal
8	08	0000 1000	BS	Backspace	Moves cursor position back one character
9	09	0000 1001	HT	Horizontal tab	Moves cursor position to next tab stop on line
10	0A	0000 1010	LF	Line feed	Moves cursor position down one line
11	0B	0000 1011	VT	Vertical tab	Moves cursor position down to next "tab line"
12	0C	0000 1100	FF	Form feed	Moves cursor position to top of next page
13	0D	0000 1101	CR	Carriage return	Moves cursor to left margin
14	0E	0000 1110	SO	Shift out	Next characters do not follow ASCII definitions
15	0F	0000 1111	SI	Shift in	Next characters revert to ASCII meaning
16	10	0001 0000	DLE	Data link escape	Used to control transmissions using "escape sequences"
17	11	0001 0001	DC1	Device control 1	Not defined; normally used for ON controls; usually user defined
18	12	0001 0010	DC2	Device control 2	Not defined; normally used for ON controls; usually user defined
19	13	0001 0011	DC3	Device control 3	Not defined; normally used for OFF controls; usually user defined
20	14	0001 0100	DC4	Device control 4	Not defined; normally used for OFF controls; usually user defined
21	15	0001 0101	NAK	Negative acknowledgment	"No" answer to questions or "errors found, retransmit"; used in asynchronous protocols
22	16	0001 0110	SYN	Synchronous idle	Sent by synchronous devices when idle to insure sync
23	17	0001 0111	ETB	End of transmission block	Marks block boundaries in transmission
24	18	0001 1000	CAN	Cancel	Indicates previous transmission should be disregarded
25	19	0001 1001	EM	End of medium	Marks end of physical media, as in paper tape
26	1A	0001 1010	SUB	Substitute	Used to replace a character known to be wrong
27	1B	0001 1011	ESC	Escape	Marks beginning of an Escape control sequence
28	1C	0001 1100	FS	File separator	Marker for major portion of transmission
29	1D	0001 1101	GS	Group separator	Marker for submajor portion of transmission
30	1E	0001 1110	RS	Record separator	Marker for minor portion of transmission
31	1F	0001 1111	US	Unit separator	Marker for most minor portion of transmission

Notes:
- ASCII control codes are sometimes used to "formalize" a communications session between communications devices.
- DC1, DC2, DC3, DC4, FS, GS, RS, and US all have user-defined meanings and may vary in use between sessions or devices.
- DC4 is often used as a general "stop transmission character."
- Codes used to control cursor position may be used to control print devices and move the print head accordingly. Not all devices support the full set of positioning codes, however.

See Also: 1.16. ASCII Character Set

1.16. ASCII Character Set

Dec	Hex	Binary	Name	Character
0	00	0000 0000	NUL	None
1	01	0000 0001	SOH	^A*
2	02	0000 0010	STX	^B*
3	03	0000 0011	ETX	^C*
4	04	0000 0100	EOT	^D*
5	05	0000 0101	ENQ	^E*
6	06	0000 0110	ACK	^F*
7	07	0000 0111	BEL	^G*
8	08	0000 1000	BS	^H*
9	09	0000 1001	HT	^I*
10	0A	0000 1010	LF	^J*
11	0B	0000 1011	VT	^K*
12	0C	0000 1100	FF	^L*
13	0D	0000 1101	CR	^M*
14	0E	0000 1110	SO	^N*
15	0F	0000 1111	SI	^O*
16	10	0001 0000	DLE	^P*
17	11	0001 0001	DC1	^Q*
18	12	0001 0010	DC2	^R*
19	13	0001 0011	DC3	^S*
20	14	0001 0100	DC4	^T*
21	15	0001 0101	NAK	^U*
22	16	0001 0110	SYN	^V*
23	17	0001 0111	ETB	^W*
24	18	0001 1000	CAN	^X*
25	19	0001 1001	EM	^Y*
26	1A	0001 1010	SUB	^Z*
27	1B	0001 1011	ESC	^[*
28	1C	0001 1100	FS	^*
29	1D	0001 1101	GS	^]*
30	1E	0001 1110	RS	^^*
31	1F	0001 1111	US	^_*
32	20	0010 0000	Space	Space
33	21	0010 0001	Exclamation point	!
34	22	0010 0010	Quotation mark	"
35	23	0010 0011	Number sign	#
36	24	0010 0100	Dollar sign	$
37	25	0010 0101	Percent sign	%
38	26	0010 0110	Ampersand	&
39	27	0010 0111	Apostrophe	'
40	28	0010 1000	Opening parenthesis	(
41	29	0010 1001	Closing parenthesis	)
42	2A	0010 1010	Asterisk	*
43	2B	0010 1011	Plus sign	+
44	2C	0010 1100	Comma	,
45	2D	0010 1101	Hyphen or minus sign	-
46	2E	0010 1110	Period	.
47	2F	0010 1111	Slash	/
48	30	0011 0000	Zero	0
49	31	0011 0001	One	1
50	32	0011 0010	Two	2
51	33	0011 0011	Three	3
52	34	0011 0100	Four	4
53	35	0011 0101	Five	5
54	36	0011 0110	Six	6
55	37	0011 0111	Seven	7
56	38	0011 1000	Eight	8
57	39	0011 1001	Nine	9
58	3A	0011 1010	Colon	:
59	3B	0011 1011	Semicolon	;
60	3C	0011 1100	Less than sign	<
61	3D	0011 1101	Equal sign	=
62	3E	0011 1110	Greater than sign	>
63	3F	0011 1111	Question mark	?

Dec	Hex	Binary	Name	Character
64	40	0100 0000	At sign	@
65	41	0100 0001	Capital A	A
66	42	0100 0010	Capital B	B
67	43	0100 0011	Capital C	C
68	44	0100 0100	Capital D	D
69	45	0100 0101	Capital E	E
70	46	0100 0110	Capital F	F
71	47	0100 0111	Capital G	G
72	48	0100 1000	Capital H	H
73	49	0100 1001	Capital I	I
74	4A	0100 1010	Capital J	J
75	4B	0100 1011	Capital K	K
76	4C	0100 1100	Capital L	L
77	4D	0100 1101	Capital M	M
78	4E	0100 1110	Capital N	N
79	4F	0100 1111	Capital O	O
80	50	0101 0000	Capital P	P
81	51	0101 0001	Capital Q	Q
82	52	0101 0010	Capital R	R
83	53	0101 0011	Capital S	S
84	54	0101 0100	Capital T	T
85	55	0101 0101	Capital U	U
86	56	0101 0110	Capital V	V
87	57	0101 0111	Capital W	W
88	58	0101 1000	Capital X	X
89	59	0101 1001	Capital Y	Y
90	5A	0101 1010	Capital Z	Z
91	5B	0101 1011	Opening bracket	[
92	5C	0101 1100	Backward slash	\
93	5D	0101 1101	Closing bracket	]
94	5E	0101 1110	Caret	^
95	5F	0101 1111	Underscore	_
96	60	0110 0000	Grave	`
97	61	0110 0001	Lowercase A	a
98	62	0110 0010	Lowercase B	b
99	63	0110 0011	Lowercase C	c
100	64	0110 0100	Lowercase D	d
101	65	0110 0101	Lowercase E	e
102	66	0110 0110	Lowercase F	f
103	67	0110 0111	Lowercase G	g
104	68	0110 1000	Lowercase H	h
105	69	0110 1001	Lowercase I	i
106	6A	0110 1010	Lowercase J	j
107	6B	0110 1011	Lowercase K	k
108	6C	0110 1100	Lowercase L	l
109	6D	0110 1101	Lowercase M	m
110	6E	0110 1110	Lowercase N	n
111	6F	0110 1111	Lowercase O	o
112	70	0111 0000	Lowercase P	p
113	71	0111 0001	Lowercase Q	q
114	72	0111 0010	Lowercase R	r
115	73	0111 0011	Lowercase S	s
116	74	0111 0100	Lowercase T	t
117	75	0111 0101	Lowercase U	u
118	76	0111 0110	Lowercase V	v
119	77	0111 0111	Lowercase W	w
120	78	0111 1000	Lowercase X	x
121	79	0111 1001	Lowercase Y	y
122	7A	0111 1010	Lowercase Z	z
123	7B	0111 1011	Opening brace	{
124	7C	0111 1100	Vertical line	|
125	7D	0111 1101	Closing brace	}
126	7E	0111 1110	Tilde	~
127	7F	0111 1111	DEL	Delete

Notes:
- Apple does not use the ASCII codes for all characters, using, for example, the lower 32 characters for graphics.
- *ASCII defines characters 0–31 to be control characters (nonprinting characters). On the Macintosh, these characters will sometimes be displayed instead of being interpreted as commands. A few software programs do use the control characters for commands, however.

See Also: 1.15. ASCII Control Codes
1.17. Apple Macintosh Character Set

1.17. Apple Macintosh Character Set

Dec	Hex	Binary	Name	Character
0	00	0000 0000	(Not defined)	
1	01	0000 0001	(Not defined)	
2	02	0000 0010	(Not defined)	
3	03	0000 0011	(Not defined)	
4	04	0000 0100	(Not defined)	
5	05	0000 0101	(Not defined)	
6	06	0000 0110	(Not defined)	
7	07	0000 0111	(Not defined)	
8	08	0000 1000	(Not defined)	
9	09	0000 1001	(Not defined)	
10	0A	0000 1010	(Not defined)	
11	0B	0000 1011	(Not defined)	
12	0C	0000 1100	(Not defined)	
13	0D	0000 1101	(Not defined)	
14	0E	0000 1110	(Not defined)	
15	0F	0000 1111	(Not defined)	
16	10	0001 0000	(Not defined)	
17	11	0001 0001	Command key *	⌘
18	12	0001 0010	Check mark *	✓
19	13	0001 0011	Filled diamond *	◆
20	14	0001 0100	Apple logo *	
21	15	0001 0101	(Not defined)	
22	16	0001 0110	(Not defined)	
23	17	0001 0111	(Not defined)	
24	18	0001 1000	(Not defined)	
25	19	0001 1001	(Not defined)	
26	1A	0001 1010	(Not defined)	
27	1B	0001 1011	(Not defined)	
28	1C	0001 1100	(Not defined)	
29	1D	0001 1101	(Not defined)	
30	1E	0001 1110	(Not defined)	
31	1F	0001 1111	(Not defined)	
32	20	0010 0000	Space	
33	21	0010 0001	Exclamation point	!
34	22	0010 0010	Quotation mark	"
35	23	0010 0011	Number sign	#
36	24	0010 0100	Dollar sign	$
37	25	0010 0101	Percent sign	%
38	26	0010 0110	Ampersand	&
39	27	0010 0111	Apostrophe	'
40	28	0010 1000	Opening parenthesis	(
41	29	0010 1001	Closing parenthesis	)
42	2A	0010 1010	Asterisk	*
43	2B	0010 1011	Plus sign	+
44	2C	0010 1100	Comma	,
45	2D	0010 1101	Hyphen or minus sign	-
46	2E	0010 1110	Period	.
47	2F	0010 1111	Slash	/
48	30	0011 0000	Zero	0
49	31	0011 0001	One	1
50	32	0011 0010	Two	2
51	33	0011 0011	Three	3
52	34	0011 0100	Four	4
53	35	0011 0101	Five	5
54	36	0011 0110	Six	6
55	37	0011 0111	Seven	7
56	38	0011 1000	Eight	8
57	39	0011 1001	Nine	9
58	3A	0011 1010	Colon	:
59	3B	0011 1011	Semicolon	;
60	3C	0011 1100	Less than sign	<
61	3D	0011 1101	Equal sign	=
62	3E	0011 1110	Greater than sign	>
63	3F	0011 1111	Question mark	?

Dec	Hex	Binary	Name	Character
64	40	0100 0000	At sign	@
65	41	0100 0001	Capital A	A
66	42	0100 0010	Capital B	B
67	43	0100 0011	Capital C	C
68	44	0100 0100	Capital D	D
69	45	0100 0101	Capital E	E
70	46	0100 0110	Capital F	F
71	47	0100 0111	Capital G	G
72	48	0100 1000	Capital H	H
73	49	0100 1001	Capital I	I
74	4A	0100 1010	Capital J	J
75	4B	0100 1011	Capital K	K
76	4C	0100 1100	Capital L	L
77	4D	0100 1101	Capital M	M
78	4E	0100 1110	Capital N	N
79	4F	0100 1111	Capital O	O
80	50	0101 0000	Capital P	P
81	51	0101 0001	Capital Q	Q
82	52	0101 0010	Capital R	R
83	53	0101 0011	Capital S	S
84	54	0101 0100	Capital T	T
85	55	0101 0101	Capital U	U
86	56	0101 0110	Capital V	V
87	57	0101 0111	Capital W	W
88	58	0101 1000	Capital X	X
89	59	0101 1001	Capital Y	Y
90	5A	0101 1010	Capital Z	Z
91	5B	0101 1011	Opening bracket	[
92	5C	0101 1100	Backward slash	\
93	5D	0101 1101	Closing bracket	]
94	5E	0101 1110	Caret	^
95	5F	0101 1111	Underscore	_
96	60	0110 0000	Grave	`
97	61	0110 0001	Lowercase a	a
98	62	0110 0010	Lowercase b	b
99	63	0110 0011	Lowercase c	c
100	64	0110 0100	Lowercase d	d
101	65	0110 0101	Lowercase e	e
102	66	0110 0110	Lowercase f	f
103	67	0110 0111	Lowercase g	g
104	68	0110 1000	Lowercase h	h
105	69	0110 1001	Lowercase i	i
106	6A	0110 1010	Lowercase j	j
107	6B	0110 1011	Lowercase k	k
108	6C	0110 1100	Lowercase l	l
109	6D	0110 1101	Lowercase m	m
110	6E	0110 1110	Lowercase n	n
111	6F	0110 1111	Lowercase o	o
112	70	0111 0000	Lowercase p	p
113	71	0111 0001	Lowercase q	q
114	72	0111 0010	Lowercase r	r
115	73	0111 0011	Lowercase s	s
116	74	0111 0100	Lowercase t	t
117	75	0111 0101	Lowercase u	u
118	76	0111 0110	Lowercase v	v
119	77	0111 0111	Lowercase w	w
120	78	0111 1000	Lowercase x	x
121	79	0111 1001	Lowercase y	y
122	7A	0111 1010	Lowercase z	z
123	7B	0111 1011	Opening brace	{
124	7C	0111 1100	Vertical line	|
125	7D	0111 1101	Closing brace	}
126	7E	0111 1110	Tilde	~
127	7F	0111 1111	(Not defined)	

(continued)

Table 1.17. Continued

Dec	Hex	Binary	Name	Character
128	80	1000 0000	A umlaut	Ä
129	81	1000 0001	A ring	Å
130	82	1000 0010	C cedilla	Ç
131	83	1000 0011	E acute	É
132	84	1000 0100	N tilde	Ñ
133	85	1000 0101	O umlaut	Ö
134	86	1000 0110	U umlaut	Ü
135	87	1000 0111	a acute	á
136	88	1000 1000	a grave	à
137	89	1000 1001	a circumflex	â
138	8A	1000 1010	a umlaut	ä
139	8B	1000 1011	a tilde	ã
140	8C	1000 1100	a ring	å
141	8D	1000 1101	c cedilla	ç
142	8E	1000 1110	e acute	é
143	8F	1000 1111	e grave	è
144	90	1001 0000	e circumflex	ê
145	91	1001 0001	e umlaut	ë
146	92	1001 0010	i acute	í
147	93	1001 0011	i grave	ì
148	94	1001 0100	i circumflex	î
149	95	1001 0101	i umlaut	ï
150	96	1001 0110	n tilde	ñ
151	97	1001 0111	o acute	ó
152	98	1001 1000	o grave	ò
153	99	1001 1001	o circumflex	ô
154	9A	1001 1010	o umlaut	ö
155	9B	1001 1011	o tilde	õ
156	9C	1001 1100	u acute	ú
157	9D	1001 1101	u grave	ù
158	9E	1001 1110	u circumflex	û
159	9F	1001 1111	u umlaut	ü
160	A0	1010 0000		†
161	A1	1010 0001	Degree symbol	°
162	A2	1010 0010	Cent sign	¢
163	A3	1010 0011	Pound sign	£
164	A4	1010 0100	Section sign	§
165	A5	1010 0101	Round bullet	•
166	A6	1010 0110	Paragraph sign	¶
167	A7	1010 0111		ß
168	A8	1010 1000	Registered mark sign	®
169	A9	1010 1001	Copyright sign	©
170	AA	1010 1010	Trademark sign	™
171	AB	1010 1011	Acute mark	´
172	AC	1010 1100	Umlaut	¨
173	AD	1010 1101	Not equal to **	≠
174	AE	1010 1110	AE ligature	Æ
175	AF	1010 1111	Large slashed zero	Ø
176	B0	1011 0000	Infinity sign **	∞
177	B1	1011 0001	Plus or minus sign **	±
178	B2	1011 0010	Less than or equal sign **	≤
179	B3	1011 0011	Greater than or equal sign **	≥
180	B4	1011 0100	Yen sign	¥
181	B5	1011 0101	Mu **	µ
182	B6	1011 0110	Delta **	∂
183	B7	1011 0111	Sigma **	∑
184	B8	1011 1000	Capital pi **	∏
185	B9	1011 1001	Small pi **	π
186	BA	1011 1010	Integral **	∫
187	BB	1011 1011	a macron	ª
188	BC	1011 1100	o macron	º
189	BD	1011 1101	Omega **	Ω
190	BE	1011 1110	ae ligature	æ
191	BF	1011 1111	Small slashed zero	ø

Dec	Hex	Binary	Name	Character
192	C0	1100 0000	Opening question mark	¿
193	C1	1100 0001	Opening exclamation	¡
194	C2	1100 0010	Upper right box **	¬
195	C3	1100 0011	Square root sign **	√
196	C4	1100 0100	Function sign **	ƒ
197	C5	1100 0101	Approximately sign **	≈
198	C6	1100 0110	Triangle **	Δ
199	C7	1100 0111	<<	«
200	C8	1100 1000	>>	»
201	C9	1100 1001	Ellipsis	...
202	CA	1100 1010	Nonbreaking space	
203	CB	1100 1011	A grave	À
204	CC	1100 1100	A tilde	Ã
205	CD	1100 1101	O tilde	Õ
206	CE	1100 1110	OE ligature	Œ
207	CF	1100 1111	oe ligature	œ
208	D0	1101 0000	En dash	–
209	D1	1101 0001	Em dash	—
210	D2	1101 0010	Opening quote mark	“
211	D3	1101 0011	Closing quote mark	”
212	D4	1101 0100	Opening single quote	‘
213	D5	1101 0101	Closing single quote	’
214	D6	1101 0110	Divide by sign **	÷
215	D7	1101 0111	Open diamond	◊
216	D8	1101 1000	y umlaut	ÿ
217	D9	1101 1001	RESERVED	
218	DA	1101 1010	RESERVED	
219	DB	1101 1011	RESERVED	
220	DC	1101 1100	RESERVED	
221	DD	1101 1101	RESERVED	
222	DE	1101 1110	RESERVED	
223	DF	1101 1111	RESERVED	
224	E0	1110 0000	RESERVED	
225	E1	1110 0001	RESERVED	
226	E2	1110 0010	RESERVED	
227	E3	1110 0011	RESERVED	
228	E4	1110 0100	RESERVED	
229	E5	1110 0101	RESERVED	
230	E6	1110 0110	RESERVED	
231	E7	1110 0111	RESERVED	
232	E8	1110 1000	RESERVED	
233	E9	1110 1001	RESERVED	
234	EA	1110 1010	RESERVED	
235	EB	1110 1011	RESERVED	
236	EC	1110 1100	RESERVED	
237	ED	1110 1101	RESERVED	
238	EE	1110 1110	RESERVED	
239	EF	1110 1111	RESERVED	
240	F0	1111 0000	RESERVED	
241	F1	1111 0001	RESERVED	
242	F2	1111 0010	RESERVED	
243	F3	1111 0011	RESERVED	
244	F4	1111 0100	RESERVED	
245	F5	1111 0101	RESERVED	
246	F6	1111 0110	RESERVED	
247	F7	1111 0111	RESERVED	
248	F8	1111 1000	RESERVED	
249	F9	1111 1001	RESERVED	
250	FA	1111 1010	RESERVED	
251	FB	1111 1011	RESERVED	
252	FC	1111 1100	RESERVED	
253	FD	1111 1101	RESERVED	
254	FE	1111 1110	RESERVED	
255	FF	1111 1111	RESERVED	

Notes:
- * Indicates characters normally only in the system font (Chicago).
- ** Indicates characters that may not be in all fonts.

Source: Inside Macintosh, page 1-221

See Also: 1.16. ASCII Character Set

1.18. Digit Positions In Common Bases

		Digit Position Value					
Base	*Name*	*6th Pos.*	*5th Pos.*	*4th Pos.*	*3rd Pos.*	*2nd Pos.*	*1st Pos.*
2	Binary	32	16	8	4	2	1
10	Decimal	100,000	10,000	1000	100	10	1
16	Hexadecimal	1,048,576	65,536	4096	256	16	1

Notes: The first digit position is the least significant.

See Also: 1.07. Decimal to Binary Number Conversion
1.08. Decimal to Hexadecimal Number Conversion

1.19. Powers Of Two

Power	*Value*	*Common Definitions and Usage*
2^1	2	(1 bit may have 2 possible values) (2 bytes = word)
2^2	4	(4 bits = nibble, BCD Digit) (4 bytes = double word)
2^3	8	(8 bits = byte, ASCII Character)
2^4	16	(16 bits = word, Near Address) (16 bytes = paragraph)
2^5	32	(32 bits = double word, Far Address)
2^6	64	
2^7	128	
2^8	256	(1 byte may have 256 possible values) (256 bytes = page)
2^9	512	
2^10	1,024	(1,024 bytes = kilobyte)
2^11	2,048	
2^12	4,096	
2^13	8,192	
2^14	16,384	
2^15	32,768	
2^16	65,536	(65,536 bytes = segment)
2^17	131,072	
2^18	262,144	
2^19	524,288	
2^20	1,048,576	(1,048,576 bytes = megabyte)
2^21	2,097,152	
2^22	4,194,304	
2^23	8,388,608	
2^24	16,777,216	
2^25	33,554,432	

Notes: 2^15 means "2 raised to the 15th power."

See Also: 1.10. Common 680x0 Data Formats
1.13. Common Memory Area Terminology

1.20. International Sort Ordering

In ASCII sort ordering, characters are sorted as follows:
Lower numbered ASCII characters appear before higher numbered ones, thus:
- All uppercase characters appear before lowercase ones.
- Characters with diacritical marks (foreign characters) come after all letters.

ASCII sort ordering treats the alphabet like this:

Sort Position	*ASCII Code*	*Character*
1	65	A
2	66	B
3	67	C
4	68	D
5	69	E
6	70	F
7	71	G
8	72	H
9	73	I
10	74	J
11	75	K
12	76	L
13	77	M
14	78	N
15	79	O
16	80	P
17	81	Q
18	82	R
19	83	S
20	84	T
21	85	U
22	86	V
23	87	W
24	88	X
25	89	Y
26	90	Z
27	97	a
28	98	b
29	99	c
30	100	d
31	101	e
32	102	f
33	103	g
34	104	h
35	105	i
36	106	j
37	107	k
38	108	l
39	109	m
40	110	n
41	111	o
42	112	p
43	113	q
44	114	r
45	115	s
46	116	t
47	117	u

Sort Position	*ASCII Code*	*Character*
48	118	v
49	119	w
50	120	x
51	121	y
52	122	z
53	128	Ä
54	129	Å
55	130	Ç
56	131	É
57	132	Ñ
58	133	Ö
59	134	Ü
60	135	á
61	136	à
62	137	â
63	138	ä
64	139	ã
65	140	å
66	141	ç
67	142	é
68	143	è
69	144	ê
70	145	ë
71	146	í
72	147	ì
73	148	î
74	149	ï
75	150	ñ
76	151	ó
77	152	ò
78	153	ô
79	154	ö
80	155	õ
81	156	ú
82	157	ù
83	158	û
84	159	ü
85	167	ß
86	174	Æ
87	190	æ
88	203	À
89	204	Ã
90	205	Õ
91	206	Œ
92	207	œ
93	216	ÿ

(continued)

Table 1.20. Continued

In international sort ordering using the Macintosh's International Utilities Package, sort order is changed as follows:
- Characters are sorted by alphanumeric order, regardless of case or diacritical marks
(i.e., case and diacritical marks are ignored).
- Ligatures (two-character combinations, like Æ) are treated differently in Germany.*

International sort ordering treats the alphabet like this:

Precedence	*Row Code*	*Characters*
1 - 31	$01 - $1F	ASCII control characters (in order)
32	$20	Space
33	$21	!
34	$22	" “ ” « »
35 - 38	$23 - $26	#, $, %, & (in that order)
39	$27	' ‘ ’
40 - 63	$28 - $3F	Remaining punctuation, in ASCII order
64	$40	@
65	$41	A Á Ă Ã Å a á à â ä ã å
66	$42	B b
67	$43	C Ç c ç
68	$44	D d
69	$45	E Ë e é è ê ë
70	$46	F f
71	$47	G g
72	$48	H h
73	$49	I i í ì î ï
74	$4A	J j
75	$4B	K k
76	$4C	L l
77	$4D	M m
78	$4E	N Ñ n ñ
79	$4F	O Ö Õ Ø o ó ò ô ö õ ø
80	$50	P p
81	$51	Q q
82	$52	R r
83	$53	S s
84	$54	T t
85	$55	U u ú ù û ü
86	$56	V v
87	$57	W w
88	$58	X x
89	$59	Y y ÿ
90	$5A	Z z
91 - 173	$5B - $AD	Remaining punctuation and misc. characters, in order
174	$AE	Æ æ Œ œ
175 +	$AF +	All remaining characters, in order of hex value

Notes: * Ligatures are normally ordered:
AE Æ OE Œ ae æ oe œ
In Germany, this is modified to:
AE Ä Æ ae ä æ OE O oe ö œ ss ß UE Ü ue ü

Source: Inside Macintosh, page I-503

1.21. Truth Tables for Logical Operations

AND

Condition 1	*Condition 2*	*Result*
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

OR

Condition 1	*Condition 2*	*Result*
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

EXCLUSIVE OR (XOR)

Condition 1	*Condition 2*	*Result*
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

NOT

Condition	*Result*
TRUE	FALSE
FALSE	TRUE

Notes: The resulting value is read by finding a row in which the conditions you are looking up are met and by reading the result in the right-most column of that row.

SECTION II

OPERATING SYSTEM SOFTWARE, RESOURCES, AND FILE FORMATS

2.001. Predefined System Packages

Resource	*ID*	*Description*	*Comments*
PACK	0	List manager package	Includes list definition procedure (LDEF 0)
PACK	2	Disk initialization package	Includes disk formatting resource (FMTR)
PACK	3	Standard file package	Includes resources (ALRT, DITL, and DLOG)
PACK	4	Floating-point arithmetic package	
PACK	5	Transcendental functions package	
PACK	6	International utilities package	
PACK	7	Binary-decimal conversion package	
PACK	12	Color picker package	Used by Mac II only
DRVR	2	.PRINT driver	
DRVR	9	.MPP driver for AppleTalk	
DRVR	10	.ATP driver for AppleTalk	
DRVR	12	Calculator DA	
DRVR	13	Alarm Clock DA	
DRVR	14	Key Caps DA	
DRVR	15	Control Panel DA	
DRVR	16	Chooser DA	
PTCH	0	Update for all ROMs	
PTCH	105	Update for version $69 ROM	Original 64K ROM
PTCH	117	Update for version $75 ROM	Macintosh Plus ROM
PTCH	630	Update for version $76 ROM	Macintosh SE ROM
PTCH	375	Update for version $78 ROM	Macintosh II ROM

Notes:
- System contains other general resources, including fonts, icons, INITs, and so on.
- Accurate as of System 4.1.

Source: Technical Introduction to the Macintosh Family, pages 77 to 79
Inside Macintosh, pages V-35, V-36

See Also: 2.002. Predefined Cursors
2.007. ROM Resources

2.002. Predefined Cursors

Cursor	*Cursor Name*	*Use*
	arrow	Selection in controls and on desktop
	iBeamCursor	Selection in text
	crossCursor	Drawing, shrinking, or stretching of graphic objects
	plusCursor	Selection in arrays
	watchCursor	Wait; operation in progress

Notes: Other cursors are used, but these are the only ones defined by Apple.

Source: Inside Macintosh, pages I-38, I-474

See Also: 2.024. CURS Resource Format
2.025. Cursor Data Masks

2.003. Standard Package IDs

ID	*Package Name*	*Constant Name*
2	Disk initialization	dskInit
3	Standard file	stdFile
4	Floating-point arithmetic	flPoint
5	Transcendental functions	trFunc
6	International utilities	intUtil
7	Binary-decimal conversion	bdConv

Notes: Contents of packages may change between versions of software.

Source: Inside Macintosh, page I-483

See Also: 2.001. Predefined System Packages

2.004. Standard Drivers and DAs

ID	*Driver Name*	*Comments*
2	.PRINT	Handles communication between Print Manager and printer
9	.MPP	AppleTalk driver
10	.ATP	AppleTalk driver
18	Control Panel	DA
28	.XPP	AppleTalk driver

Source: Inside Macintosh, page V-35

2.005. Standard Patches (PTCHs)

ID	*Purpose*
0	Updates applicable to all ROMs
105	Updates applicable to 64K ROMs
117	Updates applicable to 128K ROMs
376	Updates applicable to Mac II 256K ROMs
630	Updates applicable to Mac SE 256K ROMs

Source: Inside Macintosh, page V-36

2.006. Standard Alert Icons

Icon	*Icon Name*	*ID #*
	Note	1
	Caution	2
	Stop	0

Notes: Icon appearances changed beginning with System 4.2.

Source: Inside Macintosh, page I-420

See Also: 2.033. ICON Resource Format

2.007. ROM Resources

			ROM Type	
Type	*ID*	*Assigned Function*	*SE*	*II*
CDEF	0	Default button definition procedure	X	X
	1	Default scroll bar definition procedure	X	X
CURS	1	IBeamCursor (text selection)	X	X
	2	CrossCursor (drawing)	X	X
	3	PlusCursor (array selection)	X	X
	4	WatchCursor (wait)	X	X
DRVR	3	.Sound (sound driver)	X	X
	4	.Sony (floppy driver)	X	X
	9	.MPP (AppleTalk)	X	X
	$A	.ATP (AppleTalk)	X	X
	$28	.XPP (AppleTalk)	X	X
FONT	0	System font name	X	X
	$C	System font (Chicago 12)	X	X
	$180	Geneva font name		X
	$189	Geneva 9	X	X
	$18C	Geneva 12	X	X
	$200	Monaco font name		X
	$209	Monaco 9	X	X
KCHR	0	ASCII key mapping		X
KMAP	0	Keyboard mapping	X	X
MBDF	0	Default menu bar procedure	X	X
MDEF	0	Default menu definition procedure	X	X
NFNT	2	Chicago 12 font (4-bit)		X
	3	Chicago 12 font (8-bit)		X
	$22	Geneva 9 font (4-bit)		X
PACK	4	Floating-point arithmetic package	X	X
	5	Transcendental functions package	X	X
	7	Binary to decimal conversion package	X	X
SERD	0	Serial driver	X	X

(continued)

Table 2.007. Continued

			ROM Type	
Type	*ID*	*Assigned Function*	*SE*	*II*
WDEF	0	Default window definition function (document)	X	X
	1	Default window definition function (rounded)	X	X
cctb	0	Control color table*		X
clut	1	Color lookup table*		X
	2	Color lookup table*		X
	4	Color lookup table*		X
	8	Color lookup table*		X
	$7F	Color lookup table*		X
gama	0	Color correction table*		X
mitq	0	Internal memory requirements for MakeITable		X
snd	1	Brass horn		X
wctb	0	Window color table*		X

Notes:

- Uppercase and lowercase characters are recognized as different in resource types.
- * Mac II ROMs only.

Source: Inside Macintosh, pages V-32 to V-34, I-108

See Also:
2.002. Predefined Cursors
2.010. Resource Types
2.013. to 2.058. individual resource format tables

2.008. Resource Numbering

Numbers	*Usage*
0 to 127	System resource use only
128 to 32767	Available for application use
-1 to -16384	System resources owned by other system resources*
-16385 to -32768	RESERVED

Notes: *ID numbers for owned resources are assigned as follows:

Bit #	*Use*	*Contents*
0–4	ID #	
5–10	Owner ID #	
11–13	Type	000 = DRVR 001 = WDEF 010 = MDEF 011 = CDEF 100 = PDEF 101 = PACK 110, 111 = RESERVED
14–15	Always 1	

Source: Inside Macintosh, pages I-108 to I-110

See Also:
2.001. Predefined System Packages
2.006. Standard Alert Icons
2.011. to 2.068. individual resource format tables

2.009. Resource Attributes

Bit Numbers

7	6	5	4	3	2	1	0	*Constant Name*	*Allowable Values*
X								RESERVED	
	X							resSysHeap	1 = read into system heap, 0 = application heap
		X						resPurgeable	1 = purgeable, 0 = not purgeable
			X					resLocked	1 = locked, 0 = unlocked
				X				resProtected	1 = protected, 0 = unprotected
					X			resPreload	1 = preload, 0 = don't preload
						X		resChanged*	1 = write to resource file, 0 = no write
							X	RESERVED	

Notes: * resChanged should not be set directly.

Source: Inside Macintosh, pages I-110 to I-112

2.010. Resource Types

Type	*Function*	*See Also*	*Notes*
actb	Alert color table	2.011. actb, dctb, ictb Resource Formats	Added with System 4.1
atpl	Internal AppleTalk resource		Added with System 4.1
ALRT	Alert template	2.012. ALRT Resource Format	
ADBS	Service routine for Apple Desktop Bus		Added with System 4.1
bmap	Control Panel bitmap		
BNDL	Bundle	2.013. BNDL Resource Format	
boot	Boot blocks copy		Added with System 4.1
CACH	RAM cache code		
cctb	Control color table	2.014. cctb Resource Format	Added with System 4.1
CDEF	Control definition	2.015. CDEF Resource Format	
cdev	Control Panel device code	2.016. cdev Resource Format	
cicn	Color icon	2.019. cicn Resource Format	
clst	Chooser/Control Panel cached icon list		Added with System 4.1
clut	Color lookup table	2.020. clut Resource Format	
CNTL	Control template	2.021. CNTL Resource Format	
CODE	Code segment	2.022. CODE Resource Format	
crsr	Color cursor	2.023. crsr Resource Format	Added with System 4.1
ctab	Control Panel resource		
CURS	Cursor	2.024. CURS Resource Format	
dctb	Dialog color table	2.011. actb, dctb, ictb Resource Formats	Added with System 4.1
DITL	Dialog or alert item list	2.026. DITL Resource Format	
DLOG	Dialog template	2.027. DLOG Resource Format	
DRVR	Desk accessory or device driver		
DSAT	System startup alert table		
fctb	Font color table	2.028. fctb Resource Format	Added with System 4.1
finf	Font information		Added with System 4.1
FKEY	Function key (Command-Shift-Number)	2.029. FKEY Resource Format	
FMTR	Formatting code for floppy drives	2.003. Standard Package IDs	
FOND	Font family	2.030. FOND Resource Format	
FONT	Font	2.031. FONT Resource Format	
FREF	File reference	2.032. FREF Resource Format	
FRSV	Reserved system font IDs		
FWID	Font widths		
gama	Color correction table		Added with System 4.1
ICN#	Icon list	2.034. ICN# Resource Format	
ICON	Icon	2.033. ICON Resource Format	
ictb	Color table dialog item	2.011. actb, dctb, ictb Resource Formats	
INIT	Initialization	2.035. INIT Resource Format	
insc	Installer script		
INTL *	International *	2.036. INTL 0, 2.037. INTL 1 Resource Formats	
INT#	Number list owned by Find File		
itl0	International date and time formats	2.036. itl0 Resource Format	Added with System 4.1
itl1	International day and month names	2.037. itl1 Resource Format	Added with System 4.1
itl2	International utilities package sort hooks		Added with System 4.1
itlb	International utilities package script bundles	2.038. itlb Resource Format	Added with System 4.1
itlc	International configuration for script manager		Added with System 4.1

(continued)

Table 2.010. Continued

Type	*Function*	*See Also*	*Notes*
KCAP	Keyboard layout (used by Key Caps)		Added with System 4.1
KCHR	ASCII key mapping (software mapping)	2.039. KCHR Resource Format	Added with System 4.1
KMAP	Keyboard mapping (hardware mapping)	2.040. KMAP Resource Format	Added with System 4.1
KSWP	Keyboard script table		
LDEF	List definition procedure		
lmem	Low memory globals		Added with System 4.1
mach	Machine identification	2.041. mach Resource Format	
MBAR	Menu bar	2.042. MBAR Resource Format	
MBDF	Default menu definition procedure	2.043. MBDF Resource Format	Added with System 4.1
mcky	Mouse tracking resource		Added with System 4.1
mctb	Menu color table	2.044. mctb Resource Format	Added with System 4.1
MDEF	Menu definition	2.045. MDEF Resource Format	
MENU	Menu	2.046. MENU Resource Format	
mitq	Internal memory requirements for MakeITable		Added with System 4.1
MMAP	Mouse tracking code		Added with System 4.1
mppc	AppleTalk configuration code		Added with System 4.1
NBPC	AppleTalk bundle		Added with System 4.1
NFNT	128K ROM font	2.047. NFNT Resource Format	
nrct	Rectangle positions	2.048. nrct Resource Format	Added with System 4.1
PACK	Package		
PAT	Pattern	2.049. PAT Resource Format	
PAT#	Pattern list	2.050. PAT# Resource Format	
PDEF	Printing code		
PICT	Picture	2.051. PICT Resource Format	
pltt	Color palette	2.052. pltt Resource Format	Added with System 4.1
ppat	Pixel pattern	2.053. ppat Resource Format	Added with System 4.1
PREC	Print record		
PRER	Chooser device type	2.054. PRER Resource File Format	
PRES	Chooser device type	2.055. PRES Resource File Format	
PTCH	ROM patch code		
RDEV	Chooser device type		
ROvr	Code overriding ROM resources		
ROv#	List of ROM resources to be overriden	2.056. ROv# Resource Format	
scrn	Screen configuration	2.057. scrn Resource Format	
SERD	RAM serial driver		
SICN	Script symbol		
SIZE	Switcher/MultiFinder resource	2.058. SIZE Resource Format	
snd	Sound resource	2.059. snd 1, 2.060. snd 2 Resource Formats	Added with System 4.1
snth	Synthesizer resource	2.061. snth Resource Format	Added with System 4.1
STR	String	2.062. STR Resource Format	
STR#	String list	2.063. STR# Resource Format	
TEXT	Text	2.064. TEXT Resource Format	
VERS	Software version number	2.065. vers Resource Format	
wctb	Window color table	2.066. wctb Resource Format	Added with System 4.1
WDEF	Window definition	2.067. WDEF Resource Format	
WIND	Window	2.068. WIND Resource Format	

Notes: •Uppercase and lowercase characters are recognized as different in resource types.
•Three character resource names require a space character at the end (PAT, STR, for example).
• Where Apple has so specified, we have indicated the system version number where a resource type was added. Most, not all, of the remaining resources were in the original software definitions.
•*Obsolete.

Source: Inside Macintosh, pages I-107 to I-108, I-455, V-30 to V-32, V-127, V-327 to V-328

See Also: 2.011. to 2.068. individual resource format tables

2.011. actb, dctb, ictb Resource Formats

Size	*Name*	*Comments*
Long word	wCSeed	Always 0
Word	wCReserved	Always 0
Word	ctSize	-1 for default window colors
Word	wContentColor (Part ID)	
Word	wContentColor (Red value)	
Word	wContentColor (Green value)	
Word	wContentColor (Blue value)	
Word	wFrameColor (Part ID)	
Word	wFrameColor (Red value)	
Word	wFrameColor (Green value)	
Word	wFrameColor (Blue value)	
Word	wTextColor (Part ID)	
Word	wTextColor (Red value)	
Word	wTextColor (Green value)	
Word	wTextColor (Blue value)	
Word	wHiliteColor (Part ID)	
Word	wHiliteColor (Red value)	
Word	wHiliteColor (Green value)	
Word	wHiliteColor (Blue value)	
Word	wTitleBarColor (Part ID)	
Word	wTitleBarColor (Red value)	
Word	wTitleBarColor (Green value)	
Word	wTitleBarColor (Blue value)	

Source: Inside Macintosh, pages V-278 to V-279

See Also: 2.010. Resource Types
2.026. DITL Resource Format

2.012. ALRT Resource Format

Size	*Name*	*Comments*
8 bytes	Bounding Rectangle	
2 bytes	Resource ID of Item List	ID of accompanying DITL resource
4 bits	4th Stage *	
4 bits	3rd Stage *	
4 bits	2nd Stage *	
4 bits	1st Stage *	

Notes: * Format of stage data:

Bit	Item # - 1 of Default Item	Normally 0
Bit	Draw Alert Box	0 = no, 1 = yes
2 Bits	Sound Number	0–3

Source: Inside Macintosh, pages I-426 to I-427

See Also: 2.010. Resource Types
2.026. DITL Resource Format

2.013. BNDL Resource Format

Size	*Name*	*Comments*
4 bytes	Application Signature	4-character signature, as in MACA
2 bytes	Resource ID of Version Data	
2 bytes	Number Resource Types - 1	
4 bytes	Resource Type	These two fields repeat for each resource type
2 bytes	Number Resources of Type -1	
2 bytes	Local ID	These two fields repeat for each resource
2 bytes	Actual ID	

Notes: Normally a BNDL has two resource types: ICN# and FREF.

Source: Inside Macintosh, pages III-12 to III-13

See Also: 2.010. Resource Types
2.032. FREF Resource Format
2.034. ICN# Resource Format

2.014. cctb Resource Format

Size	*Name*	*Comments*
Long word	ccSeed	Always 0
Word	ccReserved	Always 0
Word	ctSize	3 for standard controls
Word	cFrameColor (Part ID)	
Word	cFrameColor (Red value)	
Word	cFrameColor (Green value)	
Word	cFrameColor (Blue value)	
Word	cBodyColor (Part ID)	
Word	cBodyColor (Red value)	
Word	cBodyColor (Green value)	
Word	cBodyColor (Blue value)	
Word	cTextColor (Part ID)	
Word	cTextColor (Red value)	
Word	cTextColor (Green value)	
Word	cTextColor (Blue value)	
Word	cThumbColor (Part ID)	
Word	cThumbColor (Red value)	
Word	cThumbColor (Green value)	
Word	cThumbColor (Blue value)	

Source: Inside Macintosh, pages V-218 to V-221

See Also: 2.010. Resource Types

2.015. CDEF Resource Format

The resource data for this format is simply the compiled code of the control definition function.

Source: Inside Macintosh, page I-333

See Also: 2.010. Resource Types
2.021. CNTL Resource Format

2.016. cdev Resource Format

The resource data for this format is simply the compiled code of the control panel function.

Notes: A cdev file must have one each of the following resources (each of ID # -4064): DITL, mach, nrct, ICN#, BNDL, FREF, cdev.

Source: Inside Macintosh, page V-327

See Also: 2.010. Resource Types

2.017. cdev Message Values

Value	*Message Name*	*Comments*
0	initDev	Initialization
1	hitDev	User clicked a dialog item
2	closeDev	User closed cdev or selected another
3	nulDev	Run desk accessory
4	updateDev	Update event
5	activDev	Activate event
6	deActivDev	Deactivate event
7	keyEvtDev	Keydown or autokey event
8	macDev	Check characteristics of machine
9	undoDev	Edit menu Undo
10	cutDev	Edit menu Cut
11	copyDev	Edit menu Copy
12	pasteDev	Edit menu Paste
13	clearDev	Edit menu Clear

Source: Inside Macintosh, page V-331

See Also: 2.016. cdev Resource Format
2.018. cdev Result Codes

2.018. cdev Result Codes

Value	*Result Code*	*Comments*
-1	cdevGenErr	Generic error
0	cdevMemErr	Insufficient memory
1	cdevResErr	Missing resource
3	cdevUnset	No error

Source: Inside Macintosh, page V-335

See Also: 2.016. cdev Resource Format
2.017. cdev Message Values

2.019. cicn Resource Format

Section Name	*Size*	*Item Name*	*Comments*
IconPMap	4 bytes	baseAddr	Base address of icon's pixel image
	2 bytes	rowBytes	rowBytes of image
	8 bytes	bounds	Rectangle of image
	2 bytes	pmVersion	Normally 0
	2 bytes	packType	Normally 0
	4 bytes	packSize	Normally 0
	4 bytes	hRes	$0048 0000
	4 bytes	vRes	$0048 0000
	2 bytes	pixelType	0 = chunky pixels
	2 bytes	pixelSize	Number of bits per pixel
	2 bytes	cmpCount	Normally 1
	2 bytes	cmpSize	pixelsize
	4 bytes	planeBytes	Normally 0
	4 bytes	pmTable	Normally 0
	4 bytes	RESERVED	Normally 0
IconMask	4 bytes	baseAddr	Base address of mask used when drawing icon
	2 bytes	rowBytes	rowBytes of mask image
	8 bytes	bounds	Rectangle of image
IconBMap	4 bytes	baseAddr	Base address of icon bit map when drawing to 1-bit screen
	2 bytes	rowBytes	rowBytes of icon image
	8 bytes	bounds	Rectangle of image
IconData	4 bytes	Handle Placeholder	Normally 0
MaskData	Varies	Icon's Mask Data	(size = IconMask.rowBytes*height)
BMapData	Varies	Icon's Bitmap Data	(size = IconBMap.rowBytes*height)
PMapCTab	Varies	Icon's Color Table	(size = 8 + (ColorTable.ctSize+1)*CTEntrySize)
PMapData	Varies	Icon's Image Data	(size = IconPMap.rowBytes*height)

Notes: Height = IconPMap^^.bounds.bottom - IconPMap^^.bounds.top

Source: Inside Macintosh, pages V-80 to V-81

See Also: 2.010. Resource Types
2.020. clut Resource Format

2.020. clut Resource Format

Size	*Item Name*	*Comments*		
4 bytes	ctSeed	Normally 0		
2 bytes	ctFlags	$0000 if pixMap color table, $8000 if device color table		
2 bytes	ctSize	Number of entries in color table minus 1		
8 * (ctSize+1) bytes	Table Data	Each table entry:	2 bytes	Value # *
			2 bytes	Red component value
			2 bytes	Green component value
			2 bytes	Blue component value

Notes: * Value # is pixel value associated with RGB color for pixMap tables only.

Source: Inside Macintosh, page V-81

See Also: 2.010. Resource Types

2.021. CNTL Resource Format

Size	*Name*	*Comments*
8 bytes	Bounding Rectangle	
2 bytes	Value	Current setting
2 bytes	Visible	TRUE = control is visible
2 bytes	Maximum	Maximum allowable setting
2 bytes	Minimum	Minimum allowable setting
2 bytes	procID	
4 bytes	refCon	Control's reference ID
n bytes	Title	Pascal string

Source: Inside Macintosh, pages I-332 to I-333

See Also: 2.010. Resource Types
2.015. CDEF Resource Format

2.022. CODE Resource Format

The resource data for this format is simply the compiled code.

Source: Programmer's Introduction to the Macintosh, page 55

See Also: 2.010. Resource Types

2.023. crsr Resource Format

Section Name	*Size*	*Item Name*	*Comments*
CCrsr	2 bytes	crsrType	Set to $8001
	4 bytes	srsrMap	Offset to pixMap structure for cursor from beginning of resource data
	4 bytes	crsrData	Offset to pixel data (crsr1Data) from beginning of resource data
	4 bytes	crsrXData	Should be 0
	2 bytes	crsrXValid	Should be 0
	4 bytes	crsrXHandle	Should be 0 (space for handle)
	32 bytes	crsr1Data	1-bit image bitmap for cursor
	32 bytes	crsrMask	1-bit image bitmap for cursor's mask
	4 bytes	crsrHotSpot	Vertical, horizontal coordinates of hot spot for cursor
	4 bytes	crsrXTable	Should be 0
	4 bytes	crsrID	Should be 0
PixMap	4 bytes	baseAddr	Should be 0
	2 bytes	rowBytes	rowBytes count for image
	8 bytes	bounds	Rectangle bounding image
	2 bytes	pmVersion	Should be 0
	2 bytes	packType	Should be 0
	4 bytes	packSize	Should be 0
	4 bytes	hRes	Should be $0048 0000
	4 bytes	vRes	Should be $0048 0000
	2 bytes	pixelType	0 = chunky pixels
	2 bytes	pixelSize	Number of bits per pixel in image bitmap
	2 bytes	cmpCount	Should be 1
	2 bytes	cmpSize	Set to pixelSize
	4 bytes	planeBytes	Should be 0
	4 bytes	pmTable	Offset to color lookup table from beginning of resource data
	4 bytes	pmReserved	Should be 0
Data	Varies	Pixel Data	
	Varies	Color Lookup Table	(See 2.020.)

Notes: Size of Pixel Data = offset to Color Lookup Table - Offset to Pixel Data.

Source: Inside Macintosh, pages V-78 to V-79

See Also: 2.010. Resource Types
2.020. clut Resource Format

2.024. CURS Resource Format

Size	*Name*	*Comments*
32 bytes	Cursor Data	Formatted as 16x16 grid (word = single pixel line)
32 bytes	Cursor Mask	Formatted as 16x16 grid (see 2.025.)
4 bytes	Cursor Hot Spot	Point that aligns with mouse position

Source: Inside Macintosh, pages I-146 and I-476

See Also: 2.010. Resource Types
2.025. Cursor Data Masks

2.025. Cursor Data Masks

Data	*Mask*	*Resulting Pixel*
0	1	White
1	1	Black
0	0	Same as pixel under cursor
1	0	Inversion of pixel under cursor

Source: Inside Macintosh, page I-147

See Also: 2.024. CURS Resource Format

2.026. DITL Resource Format

Size	*Name*	*Comments*
2 bytes	Number of Items	Number of items in DITL array that follows (i.e., n)
Varies	Item Arrays *	

Notes: •* Format of Item Arrays:

4 bytes	Placeholder for pointer	Normally 0
8 bytes	Rectangle bounds	
1 byte	Item type ***	
1 byte	# Bytes following	Length of data that follows
n bytes **	Varies ***	

•** Must be an even number.

•*** Varies by item type:

Item Type	*Content*	*Length*
ctrlItem	Resource ID	Two bytes
btnCtrl	Title	Pascal string
chkCtrl	Title	Pascal string
radCtrl	Title	Pascal string
resCtrl	Resource ID	Two bytes
statText	Text	Pascal string
editText	Text	Pascal string
iconItem	Resource ID	Two bytes
picItem	Resource ID	Two bytes
userItem	Nothing (length = 0)	Zero bytes

Source: Inside Macintosh, pages I-427 to I-429

See Also: 2.010. Resource Types
2.012. ALRT Resource Format
2.027. DLOG Resource Format

2.027. DLOG Resource Format

Size	*Name*	*Comments*
8 bytes	boundsRect	
2 bytes	procID	
1 byte	Visible Flag	
1 byte	IGNORED	
1 byte	goAway Flag	
1 byte	IGNORED	
4 bytes	refCon	
2 bytes	Resource ID of DITL list	
n bytes	Title	Pascal string

Source: Inside Macintosh, pages I-425 to I-426

See Also: 2.010. Resource Types
2.026. DITL Resource Format

2.028. fctb Resource Format

Size	*Name*	*Comments*
Long word	ctSeed	
Word	ctFlags	High bit = 1 for device, 0 for pixMap
Word	ctSize	Number of entries in ctTable -1
8 bytes	ctTable *	

Notes: •* ctTable defined as:

Size	*Name*	*Comments*
Word	value	Index or other value
6 bytes	rgb **	True color

•** rgb defined as:

Size	*Name*	*Comments*
Word	Red	Red component
Word	Green	Green component
Word	Blue	Blue component

Source: Inside Macintosh, pages V-107 and V-183

See Also: 2.010. Resource Types
2.047. NFNT Resource Format

2.029. FKEY Resource Format

The resource data for this format is simply the compiled code for the FKEY.

Notes: IDs range from 3 to 9, and 0.

Source: Technical Note 3, page 1

See Also: 2.010. Resource Types

2.030. FOND Resource Format

Size	*Name*	*Comments*
Word	fondFlags	Flags for family
Word	fondFamID	Family ID number
Word	fondFirst	ASCII code of first char in family
Word	fondLast	ASCII code of last char in family
Word	fondAscent	Maximum ascent for 1-point font
Word	fondDescent	Maximum descent for 1-point font
Word	fondLeading	Maximum leading for 1-point font
Word	fondWidMax	Maximum width for 1-point font
Long word	fondWTabOFF	Offset to width table
Long word	fondKernOff	Offset to kerning table
Long word	fondStylOff	Offset to style-mapping table
24 bytes	fondProperty	Style property information
6 bytes	fondIntl	RESERVED for international use
Varies	fondAssoc	Font association table
Varies	fondWidTab	Font width table
Varies	fondStylTab	Font style-mapping table
Varies	fondKerntab	Font kerning table

Source: Inside Macintosh, page IV-47

See Also: 2.010. Resource Types
2.031. FONT Resource Format
2.047. NFNT Resource Format

2.031. FONT Resource Format

Size	*Name*	*Comments*
Word	FontType	Type of font (see 3.053.)
Word	FirstChar	ASCII code of first char in family
Word	LastChar	ASCII code of last char in family
Word	WidMax	Maximum character width
Word	KernMax	Negative of maximum character kern value
Word	NDescent	Negative of descent
Word	FRectWidth	Width of font rectangle
Word	FRectHeight	Height of font rectangle
Word	OWTLoc	Offset to offset/width table
Word	Ascent	Ascent of font
Word	Descent	Descent of font
Word	Leading	Leading for font
Word	RowWords	Row width of bit image divided by 2
Varies	Bit Image	Font's bit image (size = 2 * rowWords * fRectHeight)
Varies	Location Table	Size = 2 * (lastChar - firstChar + 3)
Varies	Offset/Width Table	Size = 2 * (lastChar - firstChar + 3)
Varies	Character-Width Table	Size = 2 * (lastChar - firstChar + 3)
Varies	Image-Height Table	Size = 2 * (lastChar - firstChar + 3)

Source: Inside Macintosh, pages IV-42 to IV-47

See Also: 2.010. Resource Types
2.030. FOND Resource Format
2.047. NFNT Resource Format
3.053. Font Numbers

2.032. FREF Resource Format

Size	*Name*	*Comments*
4 bytes	File Type Signature	4-character signature, as in APPL
2 bytes	Local ID For Icon List	

Source: Inside Macintosh, page III-12

See Also: 2.010. Resource Types
2.013. BNDL Resource Format
2.034. ICN# Resource Format

2.033. ICON Resource Format

Size	*Name*	*Comments*
128 bytes	Icon Bitmap	Organized as 32 "rows" of 32-bits (long words)

Source: Inside Macintosh, page I-476

See Also: 2.010 Resource Types
2.034. ICN# Resource Format

2.034. ICN# Resource Format

Size	*Name*	*Comments*
n *128 bytes	Icon Bitmap	Organized as 32 "rows" of 32-bits (long words)
n * 128 bytes	Icon Mask	Organized as 32 "rows" of 32-bits (long words)

Notes: n = number of icons stored in icon list resource.

Source: Inside Macintosh, pages I-476, III-2 to III-12

See Also: 2.010 Resource Types
2.033. ICON Resource Format

2.035. INIT Resource Format

The resource data for this format is simply the compiled code.

Source: Technical Note 110, page 4

See Also: 2.010 Resource Types

2.036. INTL 0 or itl0 Resource Format

Size	*Name*	*US Value*	*Comments*
Byte	decimalPt	.	
Byte	thousSep	,	
Byte	listSep	;	
Byte	currSym1	$	Up to three chars allowed
Byte	currSym2		
Byte	currSym3		
Byte	currFmt	$#.##	*
Byte	dateOrder	m/d/y	0 = mdy, 1 = dmy, 2 = ymd, 3 = mdy, 4 = dym, 5 = ydm
Byte	shrtDateFmt	m/d/y	**
Byte	dateSep	/	
Byte	timeCycle	12-hour	0 = 24 hour, 255 = 12 hour
Byte	timeFmt	hh:mm	***
4 bytes	mornStr	AM	
4 bytes	eveStr	PM	
Byte	timeSep	:	
Byte	time1Suff		Trailing string for 24-hour cycle formats
Byte	time2Suff		
Byte	time3Suff		
Byte	time4Suff		
Byte	time5Suff		
Byte	time6Suff		
Byte	time7Suff		
Byte	time8Suff		
Byte	metricSys	No	$FF = metric, 0 = not metric
Word	intl0Vers		****

Notes:
- A NUL character (ASCII 0) is used to indicate "no such character" where appropriate.
- itl0 format may change in future versions.

•* Currency Formats:

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bits 3–0*	*Function*
X					1 = include leading integer zero
	X				1 = include trailing decimal zeroes
		X			1 = minus sign for negative, 0 = parentheses
			X		1 = currency symbol leads number
				X	NOT USED

•** Short Date Formats:

X					1 = include century number
	X				1 = include leading zero in month
		X			1 = include leading zero in day
			X	X	NOT USED

•*** Time Formats:

X					1 = include leading zero for hours
	X				1 = include leading zero for minutes
		X			1 = include leading zero for seconds
			X	X	NOT USED

•**** intl0 Versions:

0	U.S.
1	France
2	Britain
3	Germany
4	Italy
5	Netherlands
6	Belgium and Luxemberg
7	Sweden
8	Spain
9	Denmark
10	Portugal
11	French-speaking Canada
12	Norway
13	Israel
14	Japan
15	Australia
16	Arabia
17	Finland
18	French-speaking Switzerland
19	German-speaking Switzerland
20	Greece
21	Iceland
22	Malta
23	Cyprus
24	Turkey
25	Yugoslavia

Source: Inside Macintosh, pages I-495 to I-499, V-289 to V-291

See Also: 2.010. Resource Types
2.037. INTL 1 or itl1 Resource Format

2.037. INTL 1 or itl1 Resource Format

Size	*Name*	*Comments*
105 bytes	days	7 arrays of 15-byte strings
180 bytes	months	12 arrays of 15-byte strings
Byte	suppressDay *	0 = for day name, $FF for no day name
Byte	lngDateFmt **	0 = day name, day, month, year; $FF = day name, month, day, year
Byte	dayLeading0	$FF = use leading zero in day number
Byte	abbrLen	Number of chars to use when abbreviated date names
4 bytes	st0	See source for details on usage
4 bytes	st1	See source for details on usage
4 bytes	st2	See source for details on usage
4 bytes	st3	See source for details on usage
4 bytes	st4	See source for details on usage
Word	int11Vers	(See 2.036. for intl0 version numbers)
Word	localRtn	May be longer than 2 bytes

Notes:

- itl1 resource format subject to change in future versions.
- * If value not $0 or $FF, value is treated by bits:

Bit 0	Suppress day of month
Bit 1	Suppress day of week
Bit 2	Suppress month
Bit 3	Suppress year

- ** If value not $0 or $FF, value is treated as map:

Bits 0–1	First field ***
Bits 2–3	Second field ***
Bits 4–5	Third field ***
Bits 6–7	Fourth field ***

- *** 00 = day of month, 01 = day of week, 10 = month, 11 = year.

Source: Inside Macintosh, pages I-500 to I-501, V-288 to V-292

See Also: 2.010. Resource Types
2.036. INTL 0 or itl1 Resource Format

2.038. itlb Resource Format

Size	*Name*	*Comments*
Unsigned integer	itl0 ID Number	
Unsigned integer	itl1 ID Number	
Unsigned integer	itl2 id Number	
Unsigned integer	RESERVED	
Unsigned integer	RESERVED	
Unsigned integer	RESERVED	
Unsigned integer	Language Code	
Unsigned integer	Number/Date Codes	
Unsigned integer	KCHR ID Number	(See 2.039.)
Unsigned integer	SICN ID Number	

Notes: itlb supported in System file 4.1 and later only.

Source: Macintosh Technical Note160

See Also: 2.010. Resource Types
2.039. KCHR Resource Format
2.040. KMAP Resource Format

2.039. KCHR Resource Format

Size	*Name*	*Comments*
Integer	Version	
256 bytes	Indexes to Character Table	Maps all modifier states to a character table number
Integer	Character Table Array Count	Number of character tables in resource (n)
128*n bytes *	Character Table Arrays	A 128-byte array for each character table
Integer	DeadKey Record Count	Number of DeadKey records in resource (m)
8*m bytes **	DeadKey Records ***	

Notes:
- * n = number of character tables (up to 256 allowed).
- ** m = number of DeadKey records (up to 256 allowed).
- *** Format of DeadKey Records:

Size	*Table Number*
Byte	Virtual Keycode
Integer	Completor Record Count
Byte	Completor Character
Byte	Substituting Character
Byte	No Match Character
Byte	(Extra room)

Source: Macintosh Technical Note160

See Also: 2.010. Resource Types
2.038. itlb Resource Format
2.040. KMAP Resource Format

2.040. KMAP Resource Format

Size	*Name*	*Comments*
Integer	ID	
Integer	Version	
128 bytes	Keycode Map	See Technical Note 160 for definition (hi bit set = exception)
Integer	Exception Arrays Count	Number of arrays that follow in file
Varies	Exception Arrays *	

Notes: * Format of exception arrays:

Byte	Keycode	
1 bit	noXor, Xor	
3 bits	Fill bits	
4 bits	ADB Opcode	
String	Variable	Pascal string

Source: Macintosh Technical Note160

See Also: 2.010. Resource Types
2.038. itlb Resource Format
2.039. KCHR Resource Format

2.041. mach Resource Format

Size	*Name*	*Comments*
Word	Softmask *	Compared to ROM85 global variable
Word	Hardmask *	Compared to HwCfgFlgs

Notes: * Mask action examples:

Softmask	*Hardmask*	*Action*
0	$FFFF	Call cdev with macDev message
$FFFF	$0000	Appear on any machine
$7FFF	$0400	Appear on ADB-equipped machine
$3FFF	$0000	Appear on Macintosh II

Source: Inside Macintosh, page V-328

See Also: 2.010. Resource Types

2.042. MBAR Resource Format

Size	*Name*	*Comments*
2 bytes	Number of Menus	
Varies	Menu IDs	Resource ID of each menu (two bytes each)

Source: Inside Macintosh, page I-365

See Also: 2.010. Resource Types
2.045. MDEF Resource Format
2.046. MENU Resource Format

2.043. MBDF Resource Format

The resource data for this format is simply the compiled code for the menu bar definition procedure.

Notes: MBDF ID#0 is supplied with the System.

Source: Inside Macintosh, page V-250

See Also: 2.010. Resource Types
3.101. Menu Bar defProc Messages

2.044. mctb Resource Format

Size	*Name*
Word	Number of Entries
Varies	Menu Color Records

Notes: Menu color record format is the same as MCEntry.

Source: Inside Macintosh, page V-234

See Also: 2.010. Resource Types
3.094. MCEntry Data Structure

2.045. MDEF Resource Format

The resource data for this format is simply the compiled code for the menu.

Source: Inside Macintosh, pages I-363, V-248

See Also: 2.010. Resource Types
2.042. MBAR Resource Format
2.046. MENU Resource Format

2.046. MENU Resource Format

Size	*Name*	*Comments*
2 bytes	Menu ID	
2 bytes	Menu Width Placeholder	Initialize as $00
2 bytes	Menu Height Placeholder	Initialize as $00
2 bytes	Resource ID of Menu Def Proc	
2 bytes	*	Must be $00
4 bytes	enableFlags for Menu	ID of accompanying DITL resource
1 byte	Length of Title	This and next field are Pascal string
Varies	Title	
Varies	Menu Items **	
1 byte	End Marker	Must be a $00

Notes: •* This and previous two bytes used as placeholder for menu handle.
•** Menu item format:

Byte	Length of Item Text	This and next field are a Pascal string
Varies	Item Text	
Byte	Icon Number	0 = no icon for item
Byte	Key Code Equivalent	0 = no key equivalent
Byte	Key Code of Mark	0 = no mark
Byte	Character Style	

Source: Inside Macintosh, page I-364

See Also: 2.010. Resource Types
2.042. MBAR Resource Format
2.045. MDEF Resource Format

2.047. NFNT Resource Format

Size	*Name*	*Comments*
Word	FontType	Type of font (see 3.053.)
Word	FirstChar	ASCII code of first char in family
Word	LastChar	ASCII code of last char in family
Word	WidMax	Maximum character width
Word	KernMax	Negative of maximum character kern value
Word	NDescent	Negative of descent
Word	FRectWidth	Width of font rectangle
Word	FRectHeight	Height of font rectangle
Word	OWTLoc	Offset to offset/width table
Word	Ascent	Ascent of font
Word	Descent	Descent of font
Word	Leading	Leading for font
Word	RowWords	Row width of bit image divided by 2
Varies	Bit Image	Font's bit image (size = 2 * RowWords * FRectHeight)
Varies	Location Table	Size = 2 * (LastChar - FirstChar + 3)
Varies	Offset/Width Table	Size = 2 * (LastChar - FirstChar + 3)
Varies	Character-Width Table	Size = 2 * (LastChar - FirstChar + 3)
Varies	Image-Height Table	Size = 2 * (LastChar - FirstChar + 3)

Source: Inside Macintosh, pages IV-42 to IV-47

See Also: 2.010. Resource Types
2.030. FOND Resource Format
2.031. FONT Resource Format
3.053. Font Numbers

2.048. nrct Resource Format

Size	*Name*	*Comments*
Word	Number of Rectangles	
Varies	Rectangle Definitions	Each rectangle definition is 8 bytes

Notes: Rectangle coordinates are relative to the window they appear in for cdevs.

Source: Inside Macintosh, page V-329

See Also: 2.010. Resource Types

2.049. PAT Resource Format

Size	*Name*	*Comments*
8 bytes	Pattern	Organized as 8 x 8 matrix (each byte = one row)

Source: Inside Macintosh, page I-476

See Also: 2.010. Resource Types
2.050. PAT# Resource Format

2.050. PAT# Resource Format

Size	*Name*	*Comments*
2 bytes	Number of Patterns (n)	
n * 8 bytes	Patterns	Organized as 8 x 8 matrix (each byte = one row)

Source: Inside Macintosh, page I-476

See Also: 2.010. Resource Types
2.049. PAT Resource Format

2.051. PICT Resource Format

Size	*Name*	*Comments*
2 bytes	Picture Length (n + 10)	In bytes
8 bytes	Bounding Rectangle	
n bytes	Picture Data	Quickdraw data and PICT comments

Source: Inside Macintosh, pages I-462, I-476

See Also: 2.010. Resource Types
2.087. PICT File Format
2.088. PicComments
2.089. PICT Data Types
2.090. PICT Opcodes

2.052. pltt Resource Format

Size	*Name*	*Comments*
Word	pmEntries	Number of pmInfo entries
7 words	pmDataFields	Private fields
Varies	pmInfo	Array of ColorInfo entries (see 3.149.)

Source: Inside Macintosh, pages V-159 to V-160

See Also: 2.010. Resource Types
3.149. ColorInfo Data Structure

2.053. ppat Resource Format

Section Name	*Size*	*Item Name*	*Comments*
PixPat	2 bytes	patType	1 = full color pattern
	4 bytes	patMap	Offset to pixMap record from beginning of resource data
	4 bytes	patData	Offset to pixel data from beginning of resource data
	4 bytes	patXData	Should be 0
	2 bytes	patXValid	Should be -1
	4 bytes	patXMap	Should be 0
	8 bytes	pat1Data	1 bit pattern data
PixMap	4 bytes	baseAddr	Should be 0
	2 bytes	rowBytes	rowBytes count for image
	8 bytes	bounds	Rectangle bounding image
	2 bytes	pmVersion	Should be 0
	2 bytes	packType	Should be 0
	4 bytes	packSize	Should be 0
	4 bytes	hRes	Should be $0048 0000
	4 bytes	vRes	Should be $0048 0000
	2 bytes	pixelType	0 = chunky pixels
	2 bytes	pixelSize	Number of bits per pixel in image bitmap
	2 bytes	cmpCount	Should be 1
	2 bytes	cmpSize	Set to pixelSize
	4 bytes	planeBytes	Should be 0
	4 bytes	pmTable	Offset to color lookup table from beginning of resource data
	4 bytes	pmReserved	Should be 0
Data	Varies	Pixel Data	
	Varies	Color Lookup Table	(See 2.020.)

Notes: Size of Pixel Data = offset to Color Lookup Table - Offset to Pixel Data.

Source: Inside Macintosh, pages V-79 to V-80

See Also: 2.010. Resource Types
2.020. clut Resource Format

2.054. PRER Resource File Format

The resource data for this file format is simply the compiled code for the printer driver.

Notes: Must include these resources:

Type	*ID*	*Comment*
PACK	-4096	Device package (see source)
STR	-4096	AppleTalk type name
GNRL	-4096	AppleTalk NBP timeout/retry info
STR	-4093	Left button title
STR	-4092	Right button title
STR	-4091	Chooser string
BNDL	-	Icon information
STR	-4090	RESERVED

Source: Inside Macintosh, pages IV-216 to IV-217

See Also: 2.010. Resource Types
2.055. PRES Resource File Format

2.055. PRES Resource File Format

The resource data for this file format is simply the compiled code for the printer driver.

Source: Inside Macintosh, pages IV-216 to IV-217

See Also: 2.010. Resource Types
2.054. PRER Resource File Format

2.056. ROv# Resource Format

Size	*Name*	*Comments*
Word	ROM Version Number	
Word	Number of Resources - 1	
4 bytes	Resource Type	These two fields repeat for each resource type
Word	Resource ID	

Source: Inside Macintosh, page IV-20

See Also: 2.010. Resource Types

2.057. scrn Resource Format

Size	*Name*
Word	Number of Devices
Varies *	Device Info *

Notes: •* For each defined device:

Size	*Name*	*Comments*
Word	spDrvrHw	Slot manager hardware ID
Word	slot	Slot number
Long	dCtlDevBase	dCtlDevBase from DCE
Word	mode	Slot manager ID for screen's mode
Word	flagMask	Should be $77FE
Word	flags **	Device state
Word	colorTable	Resource ID of clut
Word	gammaTable	Resource ID of gama
Rect	global Rect	Global rectangle for device
Word	ctlCount	Number of control calls
Varies	Control Info ***	Control call information

•** Device state flags:

Bit	*Controls*	*Acceptable Values*
0	Screen type	0 = mono, 1 = color
11	Level	1 = main screen, 0 = subsidiary
15	Status	0 = inactive, 1 = active

•*** For each control call:

Size	*Name*	*Comments*
Word	csCode	Code for this control call
Word	length	Length of paramblk, in bytes
Varies	paramblk	Parameters to be passed to control call

Source: Inside Macintosh, page V-127

See Also: 2.010. Resource Types

2.058. SIZE Resource Format

Size	*Name*
Word	Flags *
Long word	Preferred size
Long word	Minimum size

Notes:
- Essential if MultiFinder is used; also compatible with Switcher.
- Size resource has ID -1.
- *Flags defined as:

Size	*Name*	*Comments*
Byte	RESERVED	
Bit	RESERVED	
Bit	getFrontClicks	1 = application wants activation clicks
Bit	onlyBackground	1 = application only runs in background
Bit	multiFinderAware	1 = application activates/deactivates windows
Bit	canBackground	1 = application wants to receive null events
Bit	RESERVED	
Bit	acceptSuspendResumeEvents	1 = application knows how to process suspend/resume events
Bit	RESERVED	

Source: Programmer's Guide to MultiFinder, pages 3-6 to 3-8
Technical Note 205

See Also: 2.010. Resource Types

2.059. Format 1 snd Resource Format

Size	*Name*	*Comments*
Word	Format of Resource	Should be $0001 for Format 1 snd
Word	Number of Modifiers/Synthesizers	
Varies	Modifier/Synthesizer Info *	
Word	Number of Sound Commands	
Varies	Sound Command Info	Each sound command is 8 bytes in length

Notes: * Each entry formatted as follows:

Size	*Name*
Word	Resource ID of Modifier/Synthesizer
Long word	Initialization Parameter

Source: Inside Macintosh, pages V-490 to V-493

See Also: 2.010. Resource Types
2.060. Format 2 snd Resource Format

2.060. Format 2 snd Resource Format

Size	*Name*	*Comments*
Word	Format of Resource	Should be $0002 for Format 2 snd
Word	Reference Count	
Word	Number of Sound Commands	
Varies	Sound Command Info	Each sound command is 8 bytes in length
Long word	Pointer to Data	
Long word	Number of Samples in Sound	
Long word	Sampling Rate	
Long word	Alternate Sampling Loop Start	
Long word	Alternate Sampling Loop End	
Word	Base Note	
Varies	Sound Samples	

Source: Inside Macintosh, pages V-493 to V-496

See Also: 2.010. Resource Types
2.059. Format 1 snd Resource Format

2.061. snth Resource Format

The resource data for this format is simply the compiled code for the synthesizer.

Source: Inside Macintosh, page V-476

See Also: 2.010. Resource Types

2.062. STR Resource Format

Size	*Name*
Byte	Length of String (n)
n bytes	String

Notes:
- Basically Pascal string format.
- Note that resource name has a space following it: "STR ".

Source: Inside Macintosh, page I-476

See Also: 2.010. Resource Types
2.063. STR# Resource Format

2.063. STR# Resource Format

Size	*Name*	*Comments*
2 bytes	Number of Strings	
n bytes	Strings	In Pascal string format (length, string)

Source: Inside Macintosh, page I-476

See Also: 2.010. Resource Types
2.062. STR Resource Format

2.064. TEXT Resource Format

Size	*Name*
4 bytes	Length of Text
n bytes	Text

Source: Inside Macintosh, page I-462

See Also: 2.010. Resource Types
2.062. STR Resource Format
2.063. STR# Resource Format

2.065. vers Resource Format

Size	*Name*	*Comments*	*Acceptable Values*
Byte	Version Number (major part)	Integer portion of version	Two BCD digits
Byte	Version Number (minor part)	Decimal portion of version	Two BCD digits
Byte	Version Type	Development stage	$20 = development, $40 = alpha, $60 = beta, $80 = released
Byte	Version Stage	User-defined stage number	
Integer	Country Code	Country code as defined in international utilities	
String	Version Message (Number)	Version number formatted as a Pascal string	Pascal string format
String	Version Message (Text)	Normally copyright notice	Pascal string format

Notes:
- Vers resource supported by Finder 6.1 and later only.
- Numbering of versions:
  - -Value to left of decimal (integer) is major release number.
  - -First digit to right of decimal is minor release number.
  - -Second digit to right of decimal is bug release number.

Source: Macintosh Technical Note189

See Also: 2.010. Resource Types

2.066. wctb Resource Format

Size	*Name*	*Comments*
Long word	wCSeed	RESERVED
Word	wCReserved	RESERVED
Word	ctSize	4 for standard windows
Word	wContentColor (Part ID)	
Word	wContentColor (Red value)	
Word	wContentColor (Green value)	
Word	wContentColor (Blue value)	
Word	wFrameColor (Part ID)	
Word	wFrameColor (Red value)	
Word	wFrameColor (Green value)	
Word	wFrameColor (Blue value)	
Word	wTextColor (Part ID)	
Word	wTextColor (Red value)	
Word	wTextColor (Green value)	
Word	wTextColor (Blue value)	
Word	wHiliteColor (Part ID)	
Word	wHiliteColor (Red value)	
Word	wHiliteColor (Green value)	
Word	wHiliteColor (Blue value)	
Word	wTitleBarColor (Part ID)	
Word	wTitleBarColor (Red value)	
Word	wTitleBarColor (Green value)	
Word	wTitleBarColor (Blue value)	

Source: Inside Macintosh, pages V-202 to V-204

See Also: 2.010. Resource Types

2.067. WDEF Resource Format

Size	*Name*	*Comments*
Varies	Branch	BRA.S to Code
Word	Flags	
4 bytes	Type	Should be "WDEF"
Word	ID Number	Should be 3
Word	Version	
Varies	Code	

Notes: * One, two, or three words, depending on how far entry point is from BRA instruction.

Source: Technical Note 110, page 4

See Also: 2.010. Resource Types

2.068. WIND Resource Format

Size	*Name*	*Comments*
8 bytes	Bounding Rectangle	
2 bytes	procID of Window	
2 bytes	Visible Flag	1 = visible, 0 = not visible
2 bytes	goAway Flag	1 = goAway box, 0 = noGoAway box
4 bytes	refCon Parameter	
n bytes	Title of Window	Pascal string format (length, text)

Source: Inside Macintosh, page I-302

See Also: 2.010. Resource Types

2.069. Resource File Format

Size	*Name*	*Item*	*Comments*
4 bytes	Resource Header	Offset to Resource Data	From file start, in bytes
4 bytes		Offset to Resource Map	From file start, in bytes
4 bytes		Length of Resource Data	In bytes
4 bytes		Length of Resource Map	In bytes
112 bytes	RESERVED		Used by System
128 bytes	User-Defined		Available for application data
4 bytes	Resource Data	Length of Data	These two fields repeat for each
Varies		Actual Data *	resource defined in file
16 bytes	Resource Map	RESERVED	Copy of resource header
4 bytes		RESERVED	Handle to next resource map to search
2 bytes		RESERVED	File reference number
2 bytes		Resource File Attributes	(See 2.009.)
2 bytes		Offset to Type List	From beginning of Resource Map, in bytes
2 bytes		Offset to Name List	From beginning of Resource Map, in bytes
2 bytes	Type List	Number of Types - 1	
4 bytes		Resource Type	These three fields repeat for each
2 bytes		Number of This Type -1	resource type (from beginning of
2 bytes		Offset to Reference List for Type	Type List, in bytes)

Additional Type List Definitions			
2 bytes	Reference List	Resource ID	These 5 fields repeat for each resource
2 bytes		Offset to Name String	From beginning of Name List, in bytes
1 byte		Resource Attributes	(See 2.009.)
3 bytes		Offset to Resource Data	From beginning of Resource Data, in bytes
4 bytes		RESERVED	Handle to resource spaceholder

Additional Reference List Definitions			
1 byte	Name List	Length of Name	These 2 fields repeat for each named resource
Varies		Resource Name	

Notes: * Actual resource data takes form as shown in Tables 2.011. to 2.068.

Source: Inside Macintosh, pages I-128 to I-131

See Also: 2.009. Resource Attributes
2.010. Resource Types
2.011. to 2.068. individual resource format tables

2.070. RMAKER Syntax

Item	*Use*	*Syntax*	*Example*	*Notes*
*	Full Line Comment	* text	* This is a comment line	
;;	End of Line Comment	othercommands ;; text	10 200 10 30 ;; bounds rect	
/QUIT	Autoquit after build	/QUIT	/QUIT	
/NOSCROLL	Don't scroll window	/NOSCROLL	/NOSCROLL	
Filename	Output file name	filename	Sample.rsrc	*
Type-creator	File type and creator	typecrea	????APPL	**
\##	Hex value	\##	\14	
TYPE	Resource Type Definition	TYPE restype [=restype]	TYPE ICON = GNRL	***
INCLUDE	Merges additional resource file	INCLUDE filename	INCLUDE MyStuff.r	
Name	Resource Name/ID	[name], ID# [(attributes)]	"Thom's Resource",256 (visible)	
Resource data	Data for ALRT	top left bottom right itemlistID stagesword	20 100 200 400 10 FFFF	
	Data for BNDL	creator ID ICN# localID resID [localID resID...] FREF localID resID [localID resID...]	THOM 0 ICN# 0 256 1 257 2 258 FREF 0 256 1 257 3 258	
	Data for CNTL	title top left bottom right flags CDEFprocID controltype min max initial	MyControl 20 100 200 400 Visible 0 0 0 100 50	
	Data for DITL	numberofitems blankline itemtype top left bottom right [text/ID/or nothing] blankline itemtype top left bottom right [text/ID/nothing]...]	1 <CR> (do not put spaces in line) Button 20 20 40 100 Cancel <CR> (do not put spaces in line) Checkbox 80 20 100 120 Check Me	**** ****
	Data for DLOG	message top left bottom right flags DLOGdefID refCon DITLID	My Dialog 10 100 200 449 Visible GoAway 0 0 10	
	Data for FREF	filetype localID	APPL 0	
	Data for GNRL	.type data [.type] [.type data]	.H 0001 0002 0003 0004 .I .H 0001	†
	Data for MENU	title menuitem [menuitem...]	Thom'sMenu First Menu Item Second Menu Item	¥
	Data for PROC	filename	CodeFile	
	Data for STR	string	Some text for you--I love \14	
	Data for STR#	numberofitems string [string...]	2 First string in list Second string in list	
	Data for WIND	title top left bottom right flags WINDdefID refCon	Your Window 40 40 250 250 Visible NoGoAway 0 0	

Notes:

- Merge resources from another resource file into the output file.
- * Output file name must be first non-comment line in resource source code file; preceding exclamation point (!) makes RMAKER merge output rsrcs with existing file instead of overwriting an existing file.
- ** Type/Creator name must be second non-comment line in resource source code file. The Type/Creator name is 8 ASCII characters: the first 4 are the type, the second 4 the creator.
- *** Valid restypes are: ALRT, BNDL, CNTL, DITL, DLOG, FREF, GNRL, MENU, PROC, STR , STR#, and WIND.
- **** Resource definitions must be delimited by a blank line, i.e., a line containing only a return character.
- † Valid .types are: .P (Pascal string), .S (string), .I (integer), .L (long integer), .H (hex integer), .R (read from file). Format for .R is: .R filename resource TYPE resource ID.
- ¥ Menu item modifiers include: /char (command-char equivalent), (- (disable line).

Source: THINK's LightspeedC 3.0 User's Manual, pages 249 to 255

See Also:
2.011. to 2.068. individual resource format tables
2.069. Resource File Format
2.071. REZ Compiler Formats

2.071. REZ Compiler Formats

ResType	*Syntax*	*Comments*

actb
(also dctb)
(also ictb)
(also wctb)
	type 'actb' {
	ctSeed;
	transIndex;
	ctSize;
	ColorSpecs {
	wContentColor,
	wFrameColor,
	wTextColor,
	wHiliteColor,
	wTitleBarColor;
	red; green; blue;
	}
	};
		
		Unsigned hex longint
		Integer
		Integer = $$Countof(ColorSpec)-1
		Array[integer,integer,integer,integer]

ALRT	type 'ALRT' {
	boundsRect;
	DITL ID;
	Stages {
	stage 4,
	stage 3,
	stage 2,
	stage 1;
	}
	};
		
		Rect
		Integer
		Array[boolean,boolean,bitstring[2]]

BNDL	type 'BNDL' {
	signature;
	version ID;
	count;
	TypeArray {
	type;
	count;
	IDArray {
	LocalID;
	ActualID;
	}
	}
	};
		
		Literal longint
		Integer
		Integer = $$Countof(TypeArray)-1
		Array[literal longint,integer,array[integer,integer]]
		
		Integer = $$Countof(IDArray)-1

cctb	type 'cctb' {
	CCSeed;
	ccReserved;
	ctSize;
	ColorSpecs {
	cFrameColor,
	cBodyColor,
	cTextColor,
	cElevatorColor;
	red; green; blue;
	}
	};
		
		Unsigned hex longint
		Integer
		Integer = $$Countof(ColorSpec)-1
		Array[integer,integer,integer,integer]

(continued)

Table 2.071. Continued

ResType	*Syntax*	*Comments*
cicn	type 'cicn' {	
	baseAddress (PixMap);	Long
	pixMap flag;	Unsigned bitstring[1] = 1
	0;	Unsigned bitstring[2] = 0
	offset to next row;	Unsigned bitstring[13]
	bitmap bounds;	Rect
	pixMap version number;	Integer
	packing format;	Integer unpacked
	pixel data size;	Unsigned longint
	h.resolution (ppi);	Unsigned hex longint
	v.resoultion (ppi);	Unsigned hex longint
	storage format;	Integer
	bits per pixel;	Integer
	components per pixel;	Integer
	bits per field;	Integer
	offset to next plane;	Unsigned longint
	offset to color table;	Unsigned longint
	RESERVED;	Long
	base address (IconMask Bitmap);	Long
	row bytes;	Integer
	bitmap bounds;	Rect
	base address (IconBMap);	Long
	row bytes;	Integer
	bitmap bounds;	Rect
	placeholder;	Long
	BMapData,	Hex string
	PMapCTab,	Hex string
	PMapData	Hex string
	};	
clut (also fctb)	type 'clut' {	
	ctSeed;	Unsigned hex longint
	transIndex;	Integer
	ctSize;	Integer = $$Countof(ColorSpec)-1
	ColorSpec {	Array[integer,integer,integer,integer]
	}	
	};	
CNTL	type 'CNTL' {	
	bounds;	Rect
	value;	Integer
	visible flag;	Byte
	fill;	Byte
	max value;	Integer
	min value;	Integer
	ProcID;	Integer
	RefCon;	Longint
	Title;	pstring
	};	

(continued)

Table 2.071. Continued

ResType	*Syntax*	*Comments*
crsr	type 'crsr' {	
	cursor type;	Hex unsigned integer
	offset to pixMap;	Unsigned longint = 96
	offset to pixel data;	Unsigned longint = 146
	expanded cursor data;	Long
	expanded data depth;	Word
	RESERVED;	Long
	1-bit cursor data;	Hex string[32]
	1-bit cursor mask;	Hex string[32]
	hot spot;	Point
	table ID;	Long
	Cursor ID;	Long
	base address (pixMap);	Long
	pixMap flag;	Unsigned bitstring[1] = 1
	0;	Unsigned bitstring[2] = 0
	offset to next row;	Unsigned bitstring[13]
	bitmap bounds;	Rect
	pixMap version number;	Integer
	packing format;	Integer
	size of pixel data;	Unsigned longint
	h.resolution (ppi);	Unsigned hex longint
	v.resolution (ppi);	Unsigned hex longint
	storage format;	Integer
	bits per pixel;	Integer
	components in pixel;	Integer
	bits per field;	Integer
	offset to next plane;	Unsigned longint
	offset to color table;	Unsigned longint
	RESERVED;	Long
	Pixel data,	Hex string
	color table	Hex string
	}.	
CURS	type 'CURS' {	
	data;	Hex string[32]
	mask;	Hex string[32]
	point;	Point
	};	
DITL	type 'DITL' {	
	array size;	Integer = $$Countof(DITLarray)-1
	DITLarray {	Varies by control type, see source
	}	
	};	

(continued)

Table 2.071. Continued

ResType	*Syntax*	*Comments*

DLOG	type 'DLOG' {
	boundsRect;
	procID;
	visible flag;
	fill;
	goaway flag;
	fill;
	refCon;
	DITL ID;
	Title;
	};
		
		Rect
		Integer
		Byte
		Byte
		Byte
		Byte
		Unsigned hex longint
		Integer
		pstring

FREF	type 'FREF' {
	file type;
	icon ID;
	filename;
	};
		
		Literal longint
		Integer
		pstring

ICON	type 'ICON' {
	icon data;
	};
		
		Hex string[128]

ICN#	type 'ICN#' {
	icon data array;
	};
		
		Array[hex string[128]]

MENU	type 'MENU' {
	menu ID;
	fill;
	menu defProc ID;
	fill;
	Enable flags;
	menu enable;
	menu title;
	menu item array {
	}
	0
	};
		
		Integer
		Word[2]
		Integer
		Word
		Unsigned hex bitstring[31]
		Boolean
		pstring
		Array[pstring,byte,char,char,unsigned bitstring[8]]
		
		Byte

MBAR	type 'MBAR' {
	menu count;
	menu array {
	}
	};
		
		Integer = $$Countof(menu arrays);
		Array[integer]

mctb	type 'mctb' {
	color table count;
	MCTBarray {
	}
	};
		
		Integer = $$Countof(MCTBarray)
		Array[integer,integer,array[int,int,int],word]

PAT	type 'PAT ' {
	pattern;
	};
		
		Hex string[8]

(continued)

Table 2.071. Continued

ResType	*Syntax*	*Comments*

PAT#	type 'PAT#' {
	count;
	PatArray {
	}
	};
		
		Integer = $$Countof(PatArray)
		Array[hex string[8]]

PICT	type 'PICT' {
	length;
	frame;
	data;
	};
		
		Integer
		Rect
		Hex string

pltt	type 'pltt' {
	table count;
	RESERVED;
	RESERVED;
	RESERVED;
	RESERVED;
	ColorInfo {
	}
	};
		
		Integer = $$Countof(ColorInfo)
		Long
		Word
		Long
		Long
		Array[int,int,int,int,int,word,long]

ppat	type 'ppat' {
	pattern type;
	offset to pixMap;
	offset to data;
	expanded pixel image;
	pattern valid flag;
	expanded pattern
	old-style pattern;
	base address (pixMap);
	pixMap flag;
	0;
	offset to next row;
	bitmap bounds;
	pixMap version number;
	packing format;
	size of pixel data;
	h.resolution (ppi);
	v.resolution (ppi);
	storage format;
	bits per pixel;
	components in pixel;
	bits per field;
	offset to next plane;
	offset to color table;
	RESERVED
	cursor data,
	color table;
	};
		
		Integer
		Unsigned longint = 28
		Unsigned longint = 78
		Long
		Word
		Long
		Hex string[8]
		Long
		Unsigned bitstring[1] = 1
		Unsigned bitstring[2] = 0
		Unsigned bitstring[13]
		Rect
		Integer
		Integer
		Unsigned longint
		Unsigned hex longint
		Unsigned hex longint
		Integer
		Integer
		Integer
		Integer
		Unsigned longint
		Unsigned longint
		Long
		Hex string
		Hex string

(continued)

Table 2.071. Continued

ResType	*Syntax*	*Comments*

SICN	type 'SICN' {
	sicnarray {
	}
	};
		Array[hex string[32]]

SIZE*	type 'SIZE' {
	save screen;
	suspend resume;
	RESERVED;
	activate/deactivate;
	RESERVED;
	size - 32K
	minimum size - 32K
	};
		
		Boolean
		Boolean
		Unsigned bitstring[2] = 0
		Boolean
		Unsigned bitstring[11] = 0
		Unsigned longint
		Unsigned longint

STR	type 'STR ' {
	string;
	};
		pstring

STR#	type 'STR#' {
	string count;
	StringArray {
	}
	};
		Integer = $$Countof(StringArray)
		Array [pstring]

WIND	type 'WIND' {
	boundsRect;
	procID
	visible flag;
	fill;
	goAway flag;
	fill;
	refCon;
	title;
	};
		
		Rect
		Integer
		Byte
		Byte
		Byte
		Byte
		Unsigned hex longint
		pstring

Notes: * The definitions shown above are consistent with the old Switcher SIZE resource; newer versions of MPW use MultiFinder names for the parts as follows:

Save Screen	Boolean
Suspend Resume Events	Boolean
Option Switch	Boolean
Background	Boolean
MultiFinderAware	Boolean
backAnd Foreground	Boolean
FrontClicks	Boolean
RESERVED	Unsigned bitstring[9] = 0
prefERRedSize	Unsigned longint
minimumSize	Unsigned longint

Source: Types.r source file in MPW version 2.0.

See Also: 2.011. to 2.068. individual resource format tables
2.069. Resource File Format
2.070. RMAKER Syntax

2.072. Font ID Numbers

ID #	*Font Name*	*Supplier*	*Comments*
0	Chicago	Apple	
1	New York	Apple	
2			
3	Geneva	Apple	
4	Monaco	Apple	
5	Venice	Apple	
6	London	Apple	
7	Athens	Apple	
8	San Francisco	Apple	
9			
10			
11	Cairo	Apple	
12	Los Angeles	Apple	
13	Zapf Dingbats	Adobe	Same as in LaserWriter+, NTX
14	Bookman Light	Adobe	
15	Helvetica Narrow	Adobe	Not the same as in the LaserWriter models
16	Palatino Roman	Adobe	
17	American Typewriter Medium	Adobe	
18	Zapf Chancery Medium Italic	Adobe	
19	Souvenir Light	Adobe	
20	Times	Adobe	
21	Helvetica	Adobe	Same as in LaserWriter
22	Courier	Adobe	Same as in LaserWriter
23	Symbol	Adobe	Same as in LaserWriter
24	Mobile	Century	
25	-		
26	Lubalin graph	Adobe	
27	Benguiat Book	Adobe	
28	Friz Quadrata	Adobe	
29	Glypha	Adobe	
30	Machine	Adobe	
31	Optima	Adobe	
32	-		
33	Avant Garde Book	Adobe	
34	New Century Schoolbook	Adobe	
35 to 74	-		
75	Szwajcarskie	Allotype Typographics	
76 to 79			
80	Zip Helvetica	Century	
	Tempora Fake	Allotype Typographics	
81	Triangle Helvetica	Century	
82	Slant Helvetica	Century	
83	Random Helvetica	Century	
84	Line Helvetica	Century	
85	Inline Helvetica	Century	
	Tempora Roman	Allotype Typographics	
86	Grid Helvetica	Century	
87	Gray Scale Helvetica	Century	
88	Fountain Helvetica	Century	
89	Diamond Helvetica	Century	
	Goudy Roman	Adobe	
90	Circle Helvetica	Century	
91	Burst Helvetica	Century	
92	Slant Times	Century	
93	Gray Scale Times	Century	
94	Zip Times	Century	
95	Burst Times	Century	
96	Inline Times	Century	
	Structure	Allotype Typographics	

(continued)

Table 2.072. Continued

ID #	*Font Name*	*Supplier*	*Comments*
97	Grid Times	Century	
	Thompson	Allotype Typographics	
98	Random Times	Century	
	Haber	Allotype Typographics	
99	Diamond Times	Century	
100	Circle Times	Century	
101	Line Times	Century	
102	Triangle Times	Century	
	Czasy Roman	Allotype Typographics	
103	Fountain Times	Century	
104 to 109	-		
110	Czasy Fake	Allotype Typographics	
111 to 127	-		
128	Acropolis		
	Goudy Newstyle	Altsys	
129	Avant Garde Book Oblique	Adobe	
	Aachen	Adobe	
	Ironworks		
	Thin Extralight Willamette	Century	
130	Avant Garde Demi	Adobe	
	Kimberley		
	Thin Devoll	Century	
131	Avant Garde Demi Oblique	Adobe	
	Freestyle	Adobe	
	Parthenon		
	Fat Extralight Willamette	Century	
132	Bookman Light Italic	Adobe	
	Revue	Adobe	
	Sirius		
	Fat Devoll	Century	
133	Bookman Demi	Adobe	
	Trisse		
	Thin Medium Willamette	Century	
134	Bookman Demi Italic	Adobe	
	University Roman	Adobe	
	Cumberland	Century	
135	Courier Oblique	Adobe	
	Carta	Adobe	
	Alexandra		
	Fat Medium Willamette	Century	
136	Courier Bold	Adobe	
	Letter Gothic Bold Italic	Adobe	
	Serotype		
	Thin Cumberland	Century	
137	Courier Bold Oblique	Adobe	
	Letter Gothic Italic	Adobe	
	New East		
	Micro Times	Century	
138	Helvetica Oblique	Adobe	
	Prestige Elite Bold Italic	Adobe	
	Patriot		
139	Helvetica Bold	Adobe	
	Prestige Elite Italic	Adobe	
	Blackjack		
	Thin Times	Century	

(continued)

Table 2.072. Continued

ID #	*Font Name*	*Supplier*	*Comments*
140	Helvetica Bold Oblique	Adobe	
	Orator Slant	Adobe	
	Light Manistee	Century	
141	Narrow Helvetica Oblique	Adobe	
	Lucida	Adobe	
	Fat Times	Century	
142	Narrow Helvetica Bold	Adobe	
	Lucida Italic	Adobe	
	Moonlight		
	Bold Manistee	Century	
143	Narrow Helvetica Bold Oblique	Adobe	
	Lucida Bold	Adobe	
	Newbaby		
	Micro Courier	Century	
144	New Century Schoolbook Italic	Adobe	
	Bodoni Bold	Adobe	
	Benatar		
	Extralight Willamette	Century	
145	New Century Schoolbook Bold	Adobe	
	Bodoni Italic	Adobe	
	Kadmos		
	Milliways		
	Thin Courier	Century	
146	New Century Schoolbook Bold Italic	Adobe	
	Bodoni Poster	Adobe	
	Devoll	Century	
147	Palatino Italic	Adobe	
	Bodoni Bold Italic	Adobe	
	Fat Courier	Century	
148	Palatino Bold	Adobe	
	Lucida Bold Italic	Adobe	
149	Palatino Bold Italic	Adobe	
	Micro Helvetica	Century	
150	Times Italic	Adobe	
	Fat Cumberland	Century	
151	Times Bold	Adobe	
	Fat Helvetica	Century	
152	Times Bold Italic	Adobe	
	News Gothic Bold	Adobe	
	Spokane	Century	
153	American Typewriter Bold	Adobe	
	News Gothic	Adobe	
	Flourish		
	Thin Helvetica	Century	
154	Benguiat Bold	Adobe	
	Cooper Black	Adobe	
	Thin Spokane	Century	
155	Friz Quadrata Bold	Adobe	
	Cooper Black Italic	Adobe	
	Fat Spokane	Century	
156	Garamond Light	Adobe	
	Thin Yukon	Century	
157	Garamond Light Italic	Adobe	
	Fat Yukon	Century	
158	Garamond Bold	Adobe	
	Thin Hudson	Century	
159	Garamond Bold Italic	Adobe	
	Fat Hudson	Century	
160	Glypha Oblique	Adobe	
	Thin Rhine	Century	

(continued)

Table 2.072. Continued

ID #	Font Name	Supplier	Comments
161	Glypha Bold	Adobe	
	Hobo	Adobe	
	Fat Rhine	Century	
162	Glypha Bold Oblique	Adobe	
	Brush Script	Adobe	
	Thin Potomac	Century	
163	Lubalin Graph Book Oblique	Adobe	
	Stencil	Adobe	
	Fat Potomac	Century	
164	Lubalin Graph Demi	Adobe	
	Tiffany Roman	Adobe	
	Shady Times	Century	
165	Lubalin Graph Demi Oblique	Adobe	
	Tiffany Demi	Adobe	
	Shadow Times	Century	
166	Optima Oblique	Adobe	
	Tiffany Heavy	Adobe	
	Fill Times	Century	
167	Optima Bold	Adobe	
	Stack Times	Century	
168	Optima Bold Oblique	Adobe	
	Futura Book	Adobe	
	Light Gray Times	Century	
169	Souvenir Light Italic	Adobe	
	Futura Book Oblique	Adobe	
	Dark Gray Times	Century	
170	Souvenir Demi	Adobe	
	Tiffany Italic	Adobe	
	Thin Bold Manistee	Century	
171	Souvenir Demi Italic	Adobe	
	Fat Bold Manistee	Century	
172	Tiffany Bold Italic	Adobe	
	Thin Light Manistee	Century	
	Jott	CassadyWare	
173	Tiffany Heavy Italic	Adobe	
	Futura Light	Adobe	
	Fat Light Manistee	Century	
	Jott Italic	CassadyWare	
174	News Gothic Oblique	Adobe	
	Fat Columbia	Century	
	Jott Bold	CassadyWare	
175	News Gothic Bold Oblique	Adobe	
	Futura Bold	Adobe	
	Thin Columbia	Century	
	Jott Bold Italic	CassadyWare	
176	Fat Bold Neosho	Century	
177	Helvetica Condensed Bold	Adobe	
	Thin Bold Neosho	Century	
178	Fat Light Neosho	Century	
179	Futura Bold Oblique	Adobe	
	Thin Light Neosho	Century	
180	Bold Neosho	Century	
	Alexandria	CassadyWare	
181	Light Neosho	Century	
	Alexandria Bold	CassadyWare	
182	Galliard Bold	Adobe	
	After Times	Century	
	Alexandria Italic	CassadyWare	
183	Galliard Bold Italic	Adobe	
	After Helvetica	Century	
	Alexandria Bold Italic	CassadyWare	

(continued)

Table 2.072. Continued

ID #	Font Name	Supplier	Comments
184	Galliard Italic	Adobe	
	Shadow Helvetica	Century	
	Campanile	CassadyWare	
185	Galliard Roman	Adobe	
	Universe Light Oblique	Adobe	
	Shady Helvetica	Century	
	Giotto	CassadyWare	
186	Goudy Bold	Adobe	
	Universe Oblique	Adobe	
	Fill Helvetica	Century	
	Giotto Bold	CassadyWare	
187	Goudy Bold Italic	Adobe	
	Universe Black Oblique	Adobe	
	Stack Helvetica	Century	
	Galileo Roman	CassadyWare	
188	Goudy Italic	Adobe	
	Universe Bold Oblique	Adobe	
	Light Grey Helvetica	Century	
	Galileo Bold	CassadyWare	
189	Universe Condensed Light Oblique	Adobe	
	Dark Gray Helvetica	Century	
	Galileo Italic	CassadyWare	
190	Helvetica Condensed Light Black	Adobe	Not the same as in LaserWriter
	Trent	Century	
	Galileo Bold Italic	CassadyWare	
191	Helvetica Condensed Bold Oblique	Adobe	Not the same as in LaserWriter
	Fat Trent	Century	
	Sans Serif Bold Condensed	CassadyWare	
192	Helvetica Condensed Light	Adobe	Not the same as in LaserWriter
	Thin Trent	Century	
	Sans Serif Bold Condensed Italic	CassadyWare	
193	Helvetica Light	Adobe	Not the same as in LaserWriter
	Thingamajigs	Century	
	Sans Serif Extra Bold Condensed	CassadyWare	
194	Korinna Bold	Adobe	
	Stone Serif	Adobe	
	Sans Serif Extra Bold Condensed	CassadyWare	
195	Korinna Kursiv Bold	Adobe	
	Stone Serif Semi-Bold	Adobe	
	Doodads	Century	
	Micro Laser	CassadyWare	
196	Korinna Kursiv Regular	Adobe	
	Stone Serif Bold	Adobe	
	Micro Italic	CassadyWare	
197	Korinna	Adobe	
	Stone Serif Light	Adobe	
	Micro Bold	CassadyWare	
198	Melior Bold	Adobe	
	Stone Serif Semi-Bold Italic	Adobe	
	Gimcracks	Century	
	Micro Bold Italic	CassadyWare	
199	Melior Bold Italic	Adobe	
	Stone Serif Bold Italic	Adobe	
	Sans Serif Shadow	CassadyWare	
200	Melior Italic	Adobe	
	Stone Sans	Adobe	
	Liberty Font Outline	Image Club	
	Frontier	Altsys	
	Goudy Oldstyle	Altsys	
	Fontographer Default	Altsys	

(continued)

Table 2.072. Continued

ID #	Font Name	Supplier	Comments
201	Melior	Adobe	
	Stone Sans Semi-Bold	Adobe	
	Rubber Stamp Gray	Image Club	
	Fat Congo	Century	
	Micro Extended		
202	New Baskerville Bold	Adobe	
	Stone Sans Bold	Adobe	
	Macintosh Font Extended	Image Club	
	Paint Brush Font	Image Club	
	Fat Styx	Century	
	Micro Extended Italic	CassadyWare	
203	New Baskerville Bold Italic	Adobe	
	Stone Sans Italic	Adobe	
	Surfstyle	Image Club	
	Fat Thames	Century	
	Micro Extended Bold	CassadyWare	
204	New Baskerville Italic	Adobe	
	Stone Sans Semi-Bold Italic	Adobe	
	Thin Congo	Century	
	Micro Extended Bold Italic	CassadyWare	
205	New Baskerville Roman	Adobe	
	Stone Sans Bold Italic	Adobe	
	Mechanical Font	Image Club	
	Thin Styx	Century	
	Dorovar	CassadyWare	
206	Trump Medieval Bold	Adobe	
	Stone Informal	Adobe	
	Mechanical Outline	Image Club	
	Thin Thames	Century	
	Dorovar Italic	CassadyWare	
207	Trump Medieval Bold Italic	Adobe	
	Stone Informal Semi-Bold	Adobe	
	Crome Font	Image Club	
	Thin Colorado	Century	
	Cyrillic Bold Italic	CassadyWare	
208	Trump Medieval Italic	Adobe	
	Stone Informal Bold	Adobe	
	Modern		
	Colorado	Century	
	Cyrillic Italic	CassadyWare	
209	Trump Medieval	Adobe	
	Stone Informal Italic	Adobe	
	Fat Colorado	Century	
	Cyrillic Bold	CassadyWare	
210	Helvetica Condensed Light Black Oblique	Adobe	
	Stone Informal Semi-Bold Italic	Adobe	
	Medium Willamette	Century	
	Sans Serif	CassadyWare	
211	Helvetica Condensed Light Oblique	Adobe	
	Stone Informal Bold Italic	Adobe	
	Sans Serif Italic	CassadyWare	
212	Helvetica Light Oblique	Adobe	
	Rhine	Century	
	Sans Serif Book	CassadyWare	
213	Universe Condensed Oblique	Adobe	
	Sans Serif Book Italic	CassadyWare	
214	Universe Condensed Bold Oblique	Adobe	
	Columbia	Century	
	Sans Serif Demi Bold	CassadyWare	
215	Sans Serif Demi Italic	CassadyWare	

(continued)

Table 2.072. Continued

ID #	Font Name	Supplier	Comments
216	Sans Serif Bold	CassadyWare	
217	Sans Serif Bold Italic	CassadyWare	
218	Futura Light Oblique	Adobe	
	Yukon	Century	
	Sans Serif Extra Bold	CassadyWare	
219	Sans Serif Extra Bold Italic	CassadyWare	
220	Thames	Century	
	Monterey	CassadyWare	
221	Monterey Medium	CassadyWare	
222	Hudson	Century	
	Monterey Bold	CassadyWare	
223	Cheltenham Roman	Adobe	
	Fat Seine	Century	
	Monterey Italic	CassadyWare	
224	Cheltenham Bold	Adobe	
	Seine	Century	
	Monterey Bold Italic	CassadyWare	
225	Cheltenham Italic	Adobe	
	Universe	Adobe	
	Thin Seine	Century	
	Calligraphy	CassadyWare	
226	Cheltenham Bold Italic	Adobe	
	Universe Black	Adobe	
	Macintosh Font	Image Club	
	Gatsby Demibold Italic	CassadyWare	
227	Franklin Gothic	Adobe	
	Universe Bold	Adobe	
	Liberty Font	Image Club	
	Gatsby Demibold	CassadyWare	
228	Franklin Gothic Demibold	Adobe	
	Universe Condensed Bold	Adobe	
	Aria Font	Image Club	
	Rubber Stamp	Image Club	
	Gatsby Light Italic	CassadyWare	
229	Franklin Gothic Heavy	Adobe	
	Universe Condensed	Adobe	
	Aria Thin	Image Club	
	Surfstyle Shadow	Image Club	
	Gatsby Light	CassadyWare	
230	Prestige Elite	Adobe	
	Universe Light	Adobe	
	Brass Plate	Image Club	
	Scanning Font	Image Club	
	Regency Script	CassadyWare	
231	Prestige Elite Bold	Adobe	
	Lynz Grey	Image Club	
	Coventry Script	CassadyWare	
232	Century Old Style Roman	Adobe	
	Lynz Font	Image Club	
	Congo	Century	
	Zephyr Script	CassadyWare	
233	Century Old Style Bold	Adobe	
	Castle Gray Font	Image Club	
	Cunei		
	Gregorian	CassadyWare	
234	Century Old Style Italic	Adobe	
	Castle Font	Image Club	
	Bullets & Boxes		
	Cyrillic	CassadyWare	

(continued)

Table 2.072. Continued

ID #	*Font Name*	*Supplier*	*Comments*
235	Letter Gothic	Adobe	
	PaintBrush Gray Font	Image Club	
	Prelude Script	CassadyWare	
236	Letter Gothic Bold	Adobe	
	Paint Brush	Image Club	
	Bodoni	CassadyWare	
237	Orator	Adobe	
	Tipe Heavy Gray Font	Image Club	
	Fat Chancery	Century	
	Prelude Bold	CassadyWare	
238	Helvetica Condensed	Adobe	
	Tipe Condensed Font	Image Club	
	Styx	Century	
	Prelude Light Slant	CassadyWare	
239	Helvetica Condensed Oblique	Adobe	
	Tipe Heavy Font	Image Club	
	Thin Chancery	Century	
	Prelude Bold Slant	CassadyWare	
240	Tipe Thin Font	Image Club	
	Thin Avant Garde	Century	
	Ritz Laser	CassadyWare	
241	Helvetica Black	Adobe	
	Fat Avant Garde	Century	
	Ritz Condensed	CassadyWare	
242	Helvetica Black Oblique	Adobe	
	Sonata	Adobe	
	Thin Bookman	Century	
	Ritz Italic	CassadyWare	
243	Fat Bookman	Century	
	Bodoni Ultra Condensed Italic	CassadyWare	
244	Park Avenue	Adobe	
	Thin Schoolbook	Century	
	Bodoni Ultra Condensed	CassadyWare	
245	Fat Schoolbook	Century	
	Right Bank	CassadyWare	
246	Thin Palatino	Century	
	Logo Font	CassadyWare	
247	Fat Palatino	Century	
	Bodoni Ultra	CassadyWare	
248	-		
249	MacPaint (icons)	Claris	
250	FullPaint (icons)	Ashton-Tate	
	Franklin Gothic Heavy Italic	Adobe	
	Thin Dingbats	Century	
	Bodoni Laser	CassadyWare	
251	Franklin Gothic Oblique	Adobe	
	Fina Font	Image Club	
	Fat Dingbats	Century	
	Bodoni Italic	CassadyWare	

(continued)

Table 2.072. Continued

ID #	*Font Name*	*Supplier*	*Comments*
252	Franklin Gothic Demi Oblique	Adobe	
	Fina Font Extended	Image Club	
	Micro Avant Garde	Century	
	Bodoni Bold	CassadyWare	
253	Sofa Font	Image Club	
	Micro Palatino	Century	
	Bodoni Bold Italic	CassadyWare	
254	MacDraw (icons)	Claris	
	Bodoni Roman	Adobe	
	Sofa Extended Font	Image Club	
	Micro Chancery	Century	
255	Micro Dingbats	Century	
256	-		

Notes: • Until late 1988, Apple reserved font numbers 1–128.
• These numbers are those assigned by the publishers. Font/DA Mover, Suitcase, and Master Juggler, which handle installation of fonts, may renumber fonts in the system file in order to avoid conflicts.

Source: "A List of PostScript Font ID Numbers," Personal Publishing, June 1988, pages 58 to 59
Desktop Publishing Type & Graphics, pages A1 to A10

See Also: 2.030. FOND Resource Format
2.031. FONT Resource Format

2.073. SysEnvirons Results

Value	*Name*	*Comments*
-5502	envVersTooBig	Call not available
-5501	envBadVers	Nonpositive version number passed
-5500	envNotPresent	SysEnvirons trap not present
0	No Error	

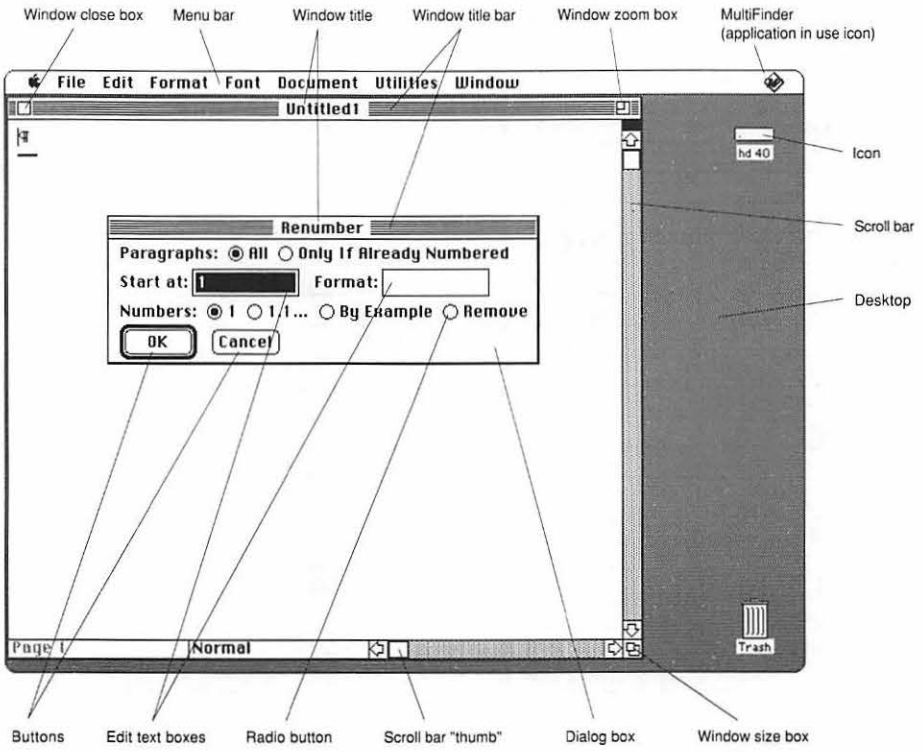
Source: Inside Macintosh, page V-11

2.074. Predefined Keys

Key	*Function*	*Comments*
Command-?	Help	
Command-B	Bold Style	Not enforced
Command-C	Copy to Clipboard	Menu equivalent
Command-Down Arrow	Open selected directory	In Standard File Package only
Command-I	Italic Style	Not enforced
Command-N	New File	Menu equivalent
Command-O	Open File	Menu equivalent
Command-P	Plain Style	Not enforced; also used for Print File
Command-Q	Quit Application	Menu equivalent
Command-S	Save File	Menu equivalent
Command-Shift-1	Eject disk in internal drive	
Command-Shift-2	Eject disk in external drive	
Command-U	Underline Style	Not enforced
Command-Up Arrow	Close current directory	In Standard File Package only
Command-V	Paste from Clipboard	Menu equivalent
Command-X	Cut to Clipboard	Menu equivalent
Command-Z	Undo Edit	Menu equivalent
Down Arrow	Scroll forward through list	In Standard File Package only
End	Scroll box to bottom and right	
Enter	Same as Return	In Standard File Package only
FwdDel	Forward delete	
Help	Help	Same as Command-?
Home	Scroll box to top and left	
PageDown	Scroll down a page	Equivalent to clicking in grey region below thumb
PageUp	Scroll up a page	Equivalent to clicking in grey region above thumb
Return	Equivalent to Open/Save button	In Standard File Package only
Tab	Equivalent to drive button	In Standard File Package only
Up Arrow	Scroll back through list	In Standard File Package only

Source: Inside Macintosh, pages IV-7, IV-74, V-22
Human Interface Guidelines, page 70

2.075. Interface Components



Source: Apple Macintosh User Interface Guidelines

2.076. Common Icons

Applications



Documents



2.077. System Software History

Release	*Intro Date*	*Intended Machines*	*System*	*Finder*	*Added*	*Changed*
3.0	Jan-86	512, 512KE, Plus	3.0	5.1		Scrapbook (2.0) ImageWriter (2.1) AppleTalk IW (2.1) LaserWriter (1.1) LaserPrep (1.1) Font/DA Mover (3.1)
No official number	Jun-86	XL, 512, 512KE, Plus	3.2	5.3		ImageWriter (2.3) AppleTalk IW (2.3) LaserWriter (3.1) LaserPrep (3.1) Font/DA Mover (3.2)
4.0	Jan-87	512KE, Plus, SE	4.0	5.4	CDEVs (Control Panel) HD Backup (no version) Disk First Aid (1.0.1)	Scrapbook (2.1) ImageWriter (2.5) AppleTalk IW (2.5) LaserWriter (3.3) LaserPrep (3.3) Font/DA Mover (3.2a3)
No official number	Apr-87	512KE, Plus, SE, II	4.1	5.5	HDSC Setup (1.4) Additional CDEVs TeachText (1.0)	Most CDEVs updated ImageWriter (2.6) AppleTalk IW (2.6) LaserWriter (4.0) LaserPrep (4.0) HD Backup (1.0) Disk First Aid (1.2) Font/DA Mover (3.5)
5.0	Sep-87	Plus, SE, II	4.2	6.0	Installer (2.5) MultiFinder (1.0) DA Handler (1.0) Backgrounder (1.0) Print Monitor (1.0) LQ ImageWriter (2.0) Apple File Exchange (1.0.1)	Most CDEVs updated HDSC Setup (1.5) TeachText (1.1) LaserWriter (5.0) LaserPrep (5.0) Disk First Aid (1.3) Font/DA Mover (3.6)
6.0.2	Oct-88	Plus, SE, II	6.0.2	6.1	MacroMaker (1.0.1) CloseView (1.0) Map (1.1) Responder (1.0)	Most CDEVs updated MultiFinder (6.0.1) DA Handler (6.0.1) Backgrounder (1.2) TeachText (1.2) ImageWriter (2.7) AppleTalk IW (2.7) LaserWriter (5.2) LaserPrep (5.2) PrintMonitor (1.2) HD Backup (1.1) Disk First Aid (1.4) Font/DA Mover (3.8) Apple File Exchange (1.1)

Notes: • Apple considers all previous releases to be obsolete and unsupported.
• Numbers in parentheses indicate version number shown in Get Info.

Source: BMUG Winter/Spring 1989 Newsletter, pages 50 to 51

2.078. Files Supplied with System

Disk	File	File Type	Version
System Tools Disk 1 (System Tools)			
	Read Me	Teach Text Document	NA
	Teach Text	Application	1.2
	System Folder	Folder	NA
	Backgrounder	System Document	1.2
	Clipboard File	System Document	NA
	Color	Control Panel Document	3.3
	DA Handler	Document	6.0
	Easy Access	Startup Document	1.0
	Finder	System Document	6.1
	General	Control Panel Document	3.3
	Key Layout	Document	2.3
	Keyboard	Control Panel Document	3.3
	Monitors	Control Panel Document	3.3
	Mouse	Control Panel Document	3.3
	MultiFinder	System Document	6.0
	Scrapbook File	System Document	NA
	Sound	Control Panel Document	3.3
	Startup Device	Control Panel Document	3.3
	System	System Document	6.0
	Setup Folder	Folder	NA
	Installer	Application	2.6
	Apple HD SC Setup	Application	1.5
	Installer Scripts	Folder	NA
	Macintosh Plus Script	Installer Document	NA
	Macintosh SE Script	Installer Document	NA
	Macintosh II Script	Installer Document	NA
System Tools Disk 2 (Printing Tools)			
	AppleTalk ImageWriter	Chooser Document	2.7
	LaserPrep	LaserWriter Document	5.2
	LaserWriter	Chooser Document	5.2
	LaserWriter IISC	Chooser Document	1.1
	ImageWriter	Chooser Document	2.6
	PrintMonitor	Application	1.2
Utility Disk 1 (Utilities 1)			
	Apple HD SC Setup	Application	1.5
	Disk First Aid	Application	1.4
	HDBackup	Application	1.1
	Installer	Application	2.6
	Special Installer Scripts	Folder	NA
	Minimum Plus Script	Installer Document	NA
	Minimum SE Script	Installer Document	NA
	Minimum II Script	Installer Document	NA
	AppleShare Script	Installer Document	NA

(continued)

Table 2.078. Continued

Disk	*File*	*File Type*	*Version*
	System Folder	Folder	NA
	Access Privileges	Font/DA Mover Document	NA
	AppleShare	Chooser Document	1.1
	Clipboard File	System Document	NA
	Color	Control Panel Document	3.3
	Easy Access	Startup Document	1.0
	Finder	System Document	6.1
	General	Control Panel Document	3.3
	Key Layout	Document	2.3
	Keyboard	Control Panel Document	3.3
	Monitors	Control Panel Document	3.3
	Mouse	Control Panel Document	3.3
	Scrapbook File	System Document	NA
	Sound	Control Panel Document	3.3
	Startup Device	Control Panel Document	3.3
	System	System Document	6.0
Utility Disk 2 (Utilities 2)			
	CloseView	Control Panel Document	1.0
	Map	Control Panel Document	1.1
	Apple File Exchange Folder	Folder	NA
	Apple File Exchange	Application	1.1
	DCA-RFT/MacWrite	Apple File Exchange Document	NA
	Font/DA Mover Folder	Folder	NA
	Font/DA Mover	Application	3.8
	Desk Accessories	Font/DA Mover Document	NA
	Fonts	Font/DA Mover Document	NA
	MacroMaker Folder	Folder	NA
	MacroMaker	Startup Document	1.0
	MacroMaker Help	MacroMaker Document	NA
	Macros	MacroMaker Document	NA

Notes: Compiled for System 6.0.0.

Source: System 6.0.0 release disks

See Also: 2.077. System Software History

2.079. File Creators and Types

Sorted by Program Name

Program	*Creator*	*Type*	*Comments*
4th Dimension	4D02	APPL	Application
4th Dimension	4D02	BAS2	
4th Dimension	4D02	data	Data file
4th Dimension	4D02	EFRM	
4th Dimension	4D02	FILT	Filter file
4th Dimension	4D02	FLAG	
4th Dimension	4D02	INDX	Index file
4th Dimension	4D02	RECH	
4th Dimension	4D02	REPT	
4th Dimension	4D02	SETT	Settings file
4th Dimension	4D02	TEXT	
4th Dimension	4D02	TFRM	
4th Dimension	4D02	VAR1	
Acta Reader	ACTA	APPL	Application
ASM (MDS)	ASM	APPL	Application
ASM (MDS)	ASM	BINF	Binary format
ASM (MDS)	ASM	PSYM	Packed symbol table
ASM (MDS)	ASM	RELB	Relocatable object
Backup	BkUp	APPL	Application
Beep Init	BeP3	APPL	Application
Calendar File	MACS	ZSYS	Data file used by System
Camera	CAM	APPL	Application
Canvas	DAD	defd	Data file
Clipboard File	MACS	CLIP	Data file used by System
Copy II Hard Disk	CPHD	APPL	Application
Copy II Mac	BCPY	APPL	Application
Cricket Draw	CRDW	APPL	Application
Cricket Graph	CGRF	APPL	Application
DA Handler	macs	dahd	
Desk Accesories	DMOV	DFIL	Data file used by Font/DA Mover
Desktop	ERIK	FNDR	Data file used by Finder
Disk Express	DEXP	APPL	Application
Disk First Aid	Scav	APPL	Application
Disk Fit	DZT4	APPL	Application
Disk Fit	DZT4	HEL4	Help file
Disk Fit	DZT4	STA4	Information file
Dollars & Sense	DAS	APPL	Application
Dollars & Sense	DAS	DASD	Data file
Double Helix	HELX	APPL	Application
Double Helix	HELX	HEAP	Formatted data
Double Helix	HELX	HEAT	
Double Helix	HELX	HEAV	
Double Helix	HELX	HEXT	Help file
Double Helix	HELX	HLOG	
Double Helix	HELX	TEXT	Text-only data file
Edit	EDIT	APPL	Application
Edit	EDIT	TEXT	Date file
Excel	XCEL	APPL	Application file
Excel	XCEL	MCBN	Chart
Excel	XCEL	TEXT	Text-only data file
Excel	XCEL	XHLP	Help

(continued)

Table 2.079. Continued

Program	*Creator*	*Type*	*Comments*
Excel	XCEL	XLBN	Worksheet
Excel	XCEL	XLIN	Resume Excel file
Excel	XCEL	XLPG	Macro file
Excel	XCEL	XLPG	Plot file
Excel	XHDI	APPL	Hard disk install
FEdit	FED2	APPL	Application (shareware version)
FEdit	FEDT	APPL	Application (registered version)
FEdit Plus	FED+	APPL	Application
FileMaker	NUTS	APPL	Application
FileMaker	NUTS	NUTD	Data file
FileMaker Plus	FMKR	APPL	Application
FileMaker Plus	FMKR	FMKD	Data file
Filevision	TELO	APPL	Application
Filevision	TELO	Help	Help file
Filevision	TELO	PICB	Data file
Finder	MACS	FNDR	System file
Finder Startup	MACS	FDOC	System file
Findswell	FIWS	APPL	Application
Font/DA Mover	DMOV	APPL	Application
Font File	DMOV	FFIL	Font file
FONTastic	FMOV	APPL	Application
Fontographer	aca2	acf2	Laser font file
Fontographer	aca2	APPL	Application
Freehand	aca3	APPL	Application
FullPaint	PANT	APPL	Application
FullPaint	PANT	PNTG	Data file
FullWrite Professional	FWRT	APPL	Application
FullWrite Professional	FWRT	FW??	Help file
FullWrite Professional	FWRT	FWDC	Main dictionary
FullWrite Professional	FWRT	FWGL	Glossary file
FullWrite Professional	FWRT	FWPR	
FullWrite Professional	FWRT	FWRT	
FullWrite Professional	FWRT	FWST	
FullWrite Professional	FWRT	FWTH	Thesaurus
FullWrite Professional	FWRT	FWUD	User dictionary
FullWrite Professional	FWRT	FWVM	
FullWrite Professional	FWRT	TEXT	Text file
Glue	SOLG	APPL	Application
Glue	SOLG	SIMA	Imagesaver file
Glue	SWRT	PRES	Print file
Hard Disk	H&R~	APPL	Runner
Hard Disk	HDUt	APPL	Utility
Hard Disk	HDUt	PARM	Utility patch file
Hard Disk	RICH	APPL	Installer
Helix	HELX	APPL	Application
Helix	HELX	HEAP	Formatted data
Helix	HELX	HEXT	Help file
Helix	HELX	TEXT	Text-only data file
HFS Backup	IGSG	APPL	
HyperCard	WILD	APPL	Application
HyperCard	WILD	STAK	Data (stack)
Illustrator 88	ARTZ	APPL	Application

(continued)

Table 2.079. Continued

Program	*Creator*	*Type*	*Comments*
ImageWriter	IWRT	PRES	
ImageWriter II	IWRT	PRES	
Installer	uins	APPL	
Keyboard	keyb	cdev	
Kolor	SSSc	cdev	
Laser Prep	LROM	PSYS	
Laser Prep	LWRT	LROM	
LaserWriter	LWRT	PRER	
LaserWriter	LWRT	PRES	
Lightspeed C	KAHL	APPL	Application
Lightspeed C	KAHL	LIB	Library file
Lightspeed C	KAHL	PROJ	Project file
Lightspeed C	KAHL	TEXT	Source code text
Link (MDS)	LINK	APPL	Application
MacCalc	MCLC	APPL	Application
MacCalc	ßHLP	ßHLP	Help file
MacDraft	MACD	APPL	Applicatoin
MacDraft	MCAD	DRWG	Drawing
MacDraw	MDRW	APPL	Application
MacDraw	MDRW	DRWG	Drawing
MacDraw	MDRW	PICT	PICT file
MacDraw II	MDPL	APPL	Application
MacLink Plus	LINK	APPL	Application
MacPaint	MPNT	APPL	Application
MacPaint	MPNT	PNTG	Painting
MacProject	MPRJ	APPL	Application
MacProject	MPRJ	MPRD	Project data file
MacroMaker	mime	INIT	Startup document
MacroMaker	mime	mcod	Macros file
MacroMaker	mime	mmab	Help file
MacsBug	MACS	ZSYS	
MacsBug (MPW)	maxb	APPL	
MacTerminal	TERM	APPL	Application
MacTerminal	TERM	TEXT	Data file
MacWrite	CABO	APPL	Rescue file
MacWrite	MACA	APPL	Application
MacWrite	MACA	TEXT	Text-only file
MacWrite	MACA	WORD	Formatted text
MacWrite II	MWII	APPL	Application
MacWrite II	MWII	STAK	Help file
MacWrite II	MWII	MW2Z	Hyphenation file
MacWrite II	CSpf	CMdt	Dictionary file
MDS Exec	EXEC	APPL	Application
Microphone	DFBO	APPL	Application
Microphone	DFBO	DFBA	Data file
Monitors	cdsc	cdev	
MultiFinder	MACS	ZSYS	Part of system
Multiplan	PLAN	APPL	Application
Multiplan	PLAN	HELP	Help
Multiplan	PLAN	MPBN	Spreadsheet
Multiplan	PLAN	TEXT	Text-only data
Omnis 3	OM$$	APPL	Application
Omnis 3	OM$$	APPL	Plus file

(continued)

Table 2.079. Continued

Program	*Creator*	*Type*	*Comments*
Omnis 3	OM$$	OM$D	Data file
Omnis 3	OM$$	OM$L	Library file
Omnis 3	OM$U	APPL	Utilities file
PackIt III	PIT	APPL	
PageMaker	ALDU	APPL	Application
PageMaker	ALDU	PBPT	Formatted data
PixelPaint	PIXR	APPL	
PowerPoint	PPNT	APPL	Application
PowerPoint	PPNT	DCT5	Dictionary
PowerPoint	PPNT	SLDS	Slides
Print Monitor	prmt	APPL	
Pyro!	Pyro	cdev	
QUED	QED1	APPL	Application
QUED	QED1	TEXT	Text file
ReadySetGo	MART	APPL	Application
ReadySetGo	MART	RSGF	Data file
Red Ryder	RRYD	APPL	Application
Red Ryder	RRYD	PDAT	Red's stuff file
Red Ryder	RRYD	RRMC	Macro file
Red Ryder	RRYD	RRRS	Data file
Reflex	RDME	APPL	Read me application
Reflex	SING	APPL	Application
Reflex	SING	SSDB	Database file
Reflex	SING	SSRF	Report file
ResEdit	RSED	APPL	
Smartcom	SCOM	APPL	Application
Smartcom II	SCOM	SCO1	Data file
Smartcom	SCOM	SCOO	Help file
Stuffit	SIT	APPL	
Suitcase	dMob	INIT	
SuperPaint	SPNT	APPL	Application
SuperPaint	SPNT	PNTG	Painting
Switcher	SWIT	APPL	Application
Switcher	SWIT	SWIT	Data file
System	LWRT	LWFN	Laserfont download file
System	MACS	ZSYS	
Think Tank	TANK	APPL	Application
Think Tank	TANK	TEXT	Data
Thunderscan	SCAN	APPL	Application
Video Works	MMPB	APPL	Player application
Video Works	MMVW	APPL	Application
Video Works	MMVW	DATA	Data
Video Works	MMVW	VWSC	Data
Word	MSWD	DICT	Dictionary
Word	MSWD	WPRD	Hyphenation
Word	WORD	APPL	Application
Word	WORD	GLOS	Glossary file
Word	WORD	PCOD	File
Word	WORD	TEXT	Text-only file
Word	WORD	WDBN	Formatted data
Word	WORD	WHLP	Help file
Word	WORD	WTMP	Temporary file
Works	PASO	TEXT	Help

(continued)

Table 2.079. Continued

Program	Creator	Type	Comments
Works	PSIP	APPL	Application
Works	PSIP	AWDT	Resume
Works	PSIP	AWSS	Spreadsheet
Works	PSIP	AWWP	Formatted text
Works	PSIP	TEXT	Text-only data
Write Now	nX^n	APPL	Application
Write Now	nX^n	APPL	Dictionary
Write Now	nX^n	nX^d	Data
Write Now	nX^n	nX^w	Data
Write Now	nX^t	APPL	Translator

Sorted by Creator

Creator	Type	Program
4D02	APPL	4th Dimension
aca2	acf2	Fontographer
aca3	APPL	Freehand
ACTA	APPL	Acta Reader
ALDU	APPL	PageMaker
ARTZ	APPL	Illustrator 88
ASM	APPL	ASM (MDS)
BCPY	APPL	Copy II Mac
BeP3	APPL	Beep Init
BkUp	APPL	Backup
CABO	APPL	MacWrite Rescue
CAM	APPL	Camera
cdsc	cdev	Monitors
CGRF	APPL	Cricket Graph
CPHD	APPL	Copy II Hard Disk
CRDW	APPL	Cricket Draw
CSpf	CMdt	MacWrite II
DAD	defd	Canvas
DAS	APPL	Dollars & Sense
DEXP	APPL	Disk Express
DFBO	APPL	Microphone
dMob	INIT	Suitcase
DMOV	APPL	Font/DA Mover
DMOV	DFIL	Desk Accesories
DMOV	FFIL	Font File
DZT4	APPL	Disk Fit
EDIT	APPL	Edit
ERIK	FNDR	Desktop
EXEC	APPL	MDS Exec
FED+	APPL	FEdit Plus
FED2	APPL	FEdit
FEDT	APPL	FEdit
FIWS	APPL	Findswell
FMKR	APPL	FileMaker Plus
FMOV	APPL	FONTastic
FWRT	APPL	FullWrite Professional
H&R~	APPL	Hard Disk
HDUt	APPL	Hard Disk
HELX	APPL	Double Helix
HELX	APPL	Helix
IGSG	APPL	HFS Backup

(continued)

Table 2.079. Continued

Creator	*Type*	*Program*
IWRT	PRES	ImageWriter
IWRT	PRES	ImageWriter II
KAHL	APPL	Lightspeed C
keyb	cdev	Keyboard
LINK	APPL	Link (MDS)
LINK	APPL	MacLink Plus
LROM	PSYS	Laser Prep
LWRT	LROM	Laser Prep
LWRT	LWFN	System
LWRT	PRER	LaserWriter
MACA	APPL	MacWrite
MACD	APPL	MacDraft
MACS	CLIP	Clipboard File
macs	dahd	DA Handler
MACS	FDOC	Finder Startup
MACS	FNDR	Finder
MACS	ZSYS	Calendar File
MACS	ZSYS	MacsBug
MACS	ZSYS	MultiFinder
MACS	ZSYS	System
MART	APPL	ReadySetGo
maxb	APPL	MacsBug (MPW)
MCAD	DRWG	MacDraft
MCLC	APPL	MacCalc
MDPL	APPL	MacDraw II
MDRW	APPL	MacDraw
mime	INIT	MacroMaker
MMPB	APPL	Video Works
MMVW	APPL	Video Works
MPNT	APPL	MacPaint
MPRJ	APPL	MacProject
MSWD	DICT	Word
MWII	APPL	MacWrite II
NUTS	APPL	FileMaker
nX^n	APPL	Write Now
nX^t	APPL	Write Now
OM$$	APPL	Omnis 3
OM$U	APPL	Omnis 3
PANT	APPL	FullPaint
PASO	TEXT	Works
PIT	APPL	PackIt III
PIXR	APPL	PixelPaint
PLAN	APPL	Multiplan
PPNT	SLDS	PowerPoint
prmt	APPL	Print Monitor
PSIP	APPL	Works
Pyro	cdev	Pyro!
QED1	APPL	QUED
RDME	APPL	Reflex
RICH	APPL	Hard Disk
RRYD	APPL	Red Ryder
RSED	APPL	ResEdit
SCAN	APPL	Thunderscan

(continued)

Table 2.079. Continued

Creator	*Type*	*Program*
Scav	APPL	Disk First Aid
SCOM	APPL	Smartcom
SING	APPL	Reflex
SIT!	APPL	Stuffit
SOLG	APPL	Glue
SPNT	APPL	SuperPaint
SSSc	cdev	Kolor
SWIT	APPL	Switcher
SWRT	PRES	Glue
TANK	APPL	Think Tank
TELO	APPL	Filevision
TERM	APPL	MacTerminal
uins	APPL	Installer
WILD	APPL	HyperCard
WORD	APPL	Word
XCEL	APPL	Excel
XHDI	APPL	Excel
ßHLP	ßHLP	MacCalc

Sorted by File Type

Type	*Function*
acf2	Fontographer laserfont file
APPL	Application
AWDT	Works resume file
AWSS	Works spreadsheet file
AWWP	Works formatted text file
BAS2	4th Dimension file
BINF	Binary format
cdev	cdev (control panel device)
CLIP	System file (clipboard)
CMdt	MacWrite II dictionary file
dahd	System file (DA handler)
DASD	Dollars and Sense data file
data	4th Dimension data file
DATA	VideoWorks data file
DCT5	PowerPoint dictionary file
defd	Canvas data file
DFBA	Microphone data file
DFIL	Font/DA Move DA file
DRWG	Drawing file
EFRM	4th Dimension file
FDOC	Finder startup file
FFIL	Font file
FILT	Filter file
FLAG	
FMKD	FileMaker data file
FNDR	Data file used by Finder
FW??	FullWrite help file
FWDC	FullWrite dictionary file
FWGL	FullWrite glossary file
FWPR	FullWrite file
FWRT	FullWrite file

(continued)

Table 2.079. Continued

Type	*Function*
FWST	FullWrite file
FWTH	FullWrite thesaurus file
FWUD	FullWrite user dictionary file
FWVM	FullWrite file
GLOS	Glossary file
HEAP	Helix formatted data
HEAT	Helix file
HEAV	Helix file
HEL4	Help file
HELP	Help file
Help	Help file
HEXT	Helix help file
HLOG	Helix file
INDX	Index file
INIT	INIT file
LIB	Library file
LROM	LaserWriter file
LWFN	Laserfont download file
MCBN	Chart
mcod	Macros file
mmab	Help file
MPBN	Spreadsheet
MPRD	Project data file
MW2Z	MacWrite II hyphenation file
NUTD	Data file
nX^d	Data
nX^w	Data
OM$D	Data file
OM$L	Library file
PARM	Utility patch file
PBPT	Formatted data
PCOD	File
PDAT	Red's stuff file
PICB	Data file
PICT	PICT file
PNTG	Painting
PRER	Printer file
PRES	Printer file
PROJ	Project file
PSYM	Packed symbol table
PSYS	Printer system file
RECH	4th Dimension file
RELB	Relocatable object
REPT	4th Dimension file
RRMC	Macro file
RRRS	Data file
RSGF	Data file
SCO1	Data file
SCOO	Help file
SETT	Settings file
SIMA	Imagesaver file
SLDS	Slides

(continued)

Table 2.079. Continued

Type	*Function*
SSDB	Database file
SSRF	Report file
STA4	Information file
STAK	Data (stack)
SWIT	Data file
TEXT	Text file (may be formatted)
TFRM	4th Dimension file
VAR1	4th Dimension file
VWSC	Data
WDBN	Formatted data
WHLP	Help file
WORD	Formatted text
WPRD	Hyphenation
WTMP	Temporary file
XHLP	Help
XLBN	Worksheet
XLIN	Resume Excel file
XLPG	Macro file
XLPG	Plot file
ZSYS	Data file used by System
ßHLP	Help file

Source: 1st Aid Kit Manual, pages 263 to 281
Program Source Disks

See Also: 2.078. Files Supplied with System

2.080. System Error Messages

ID	*Description*	*Comments*
1	Bus Error	Only Mac II and Mac XL (Lisa); invalid memory reference on bus
2	Address Error	Usually an odd-address reference
3	Illegal Instruction	Illegal 680x0 instruction encountered
4	Zero Divide	DIVS or DIVU encountered 0 divisor
5	Check Exception	CHK execution failed, etc. *
6	TrapV Exception	TRAPV execution failed, etc. *
7	Privilege Violation	Erroneous RTE executed, etc. *
8	Trace Exception	Trace bit in 680x0 status register is set
9	Line 1010 Exception	Trap dispatcher failed
10	Line 1111 Exception	Unimplemented instruction
11	Miscellaneous Exception	Exception not defined above
12	Unimplemented Routine	Attempted to execute unimplemented trap instruction
13	Spurious Interrupt	Attempted to execute NIL interrupt, etc. *
14	I/O System Error	Either File Manager or Device Manager encountered error, etc. *
15	Segment Loader Error	Segment couldn't be read into memory
16	Floating-Point Error	Halt bit in the floating point environment word is set
17–24	Can't Load Package	Package couldn't be loaded into memory, etc. *
25	Can't Allocate Memory	Memory allocation couldn't be made in heap
26	Segment Loader Error	Call to CODE 0 resource failed; unexecutable file
27	File Map Destroyed	The block number is outside range for volume
28	Stack Overflow Error	Stack expanded into the heap
30	Insert Disk	File Manager alert for disk swapping
41	Can't Find Finder	Finder wasn't found on disk
100	Can't Mount Startup Volume	The system resource file coudn't be read
32767	Sorry, System Error	Default error message

Notes: • Accurate as of System 4.1.
•* Other meanings may apply; see source for more information on possible error conditions.

Source: Technical Introduction to the Macintosh Family, pages 189 to 190

2.081. System Software Result Codes

Value	*Name*	*Type*	*Description*
0	noErr	-	No error
1	evtNotEnb	Event Manager	Event type is not one in system event mask
128	iPrAbort	Printing Manager	Abort was requested by application or user
-1	iPrSavPFil	Printing Manager	Spool file being saved
-1	qErr	Queuing Error	Entry not in the queue
-2	vTypErr	Queuing Error	QType field in vertical retrace queue isn't 'vType
-17	controlErr	Device Manager	Driver can't respond to control call, or unimplemented control instruction (Print Mgr)
-18	statusErr	Device Manager	Driver can't respond to Status call
-19	readErr	Device Manager	Driver can't respond to Read call
-20	writErr	Device Manager	Driver can't respond to Write call
-21	badUnitErr	Device Manager	Unit table and reference number mismatch
-22	unitEmptyErr	Device Manager	Reference to NIL handle in unit table
-23	openErr	Device Manager	Read/Write permission denied, or RAM serial driver open failed
-25	dRemovErr	Device Manager	Can't remove an open driver
-26	dInstErr	Device Manager	Driver not in resource file
-27	abortErr	Device Manager	I/O request aborted by KillIO call
-28	notOpenErr	Device Manager	Driver isn't open
-33	dirFulErr	File Manager	Directory is full
-34	dskFulErr	File Manager	Disk is full
-35	nsvErr	File Manager	Volume doesn't exist
-36	ioErr	File Manager	I/O error occurred
-37	badNamErr	File Manager	Bad file or volume name (check for 0 length)
-38	fnOpnErr	File Manager	File isn't open
-39	eofErr	File Manager	End-of-file encountered during read
-40	posErr	File Manager	Encountered attempt to position before start-of-file
-42	tmfoErr	File Manager	Too many files are open
-43	fnfErr	File Manager	File not found
-44	wPrErr	File Manager	Hardware lock for volume
-45	fLckdErr	File Manager	File is locked
-46	vLckdErr	File Manager	Software lock for volume
-47	fBsyErr	File Manager	File busy
-48	dupFNErr	File Manager	Name and version number already exists
-49	opWrErr	File Manager	A write access path to file already exists
-50	paramErr	File Manager	Parameter list error (see source)
-51	rfNumErr	File Manager	Reference number refers to nonexistent access path
-52	gfpErr	File Manager	Error during GetFPos
-53	volOffLinErr	File Manager	Volume requested is not on-line
-54	permErr	File Manager	Request to open locked file for writing (illegal)
-55	volOnLinErr	File Manager	Volume requested is already mounted and on-line
-56	nsDrvErr	File Manager	No such drive
-57	noMacDskErr	File Manager	Not a Macintosh format disk
-58	extFSErr	File Manager	External file system error
-59	fsRnErr	File Manager	Error during rename operation
-60	badMDBErr	File Manager	Master directory block is bad; reinitialize volume
-61	wrPermErr	File Manager	Writing to file not allowed
-64	noDriveErr	Disk Error (low level)	No drive found
-65	offLinErr	Disk Error (low level)	No disk is in the drive
-66	noNybErr	Disk Error (low level)	No data read from disk (disk probably blank)
-67	noAdrMkErr	Disk Error (low level)	Address mark wasn't found
-68	dataVerErr	Disk Error (low level)	Verify failed
-69	badCksmErr	Disk Error (low level)	Address mark is bad
-70	badBtSlpErr	Disk Error (low level)	Address mark is bad
-71	noDtaMkErr	Disk Error (low level)	Data mark wasn't found
-72	badDCksum	Disk Error (low level)	Data mark is bad
-73	badDBtSlp	Disk Error (low level)	Data mark is bad
-74	wrUnderrun	Disk Error (low level)	Write underrun
-75	cantStepErr	Disk Error (low level)	Drive error, couldn't step
-76	tk0BadErr	Disk Error (low level)	Track 0 was not found
-77	initIWMErr	Disk Error (low level)	Controller chip couldn't be initialized
-78	twoSideErr	Disk Error (low level)	Tried to read second side of single-sided disk
-79	spdAdjErr	Disk Error (low level)	Speed couldn't be adjusted
-80	seekErr	Disk Error (low level)	Other type of drive error
-81	sectNFErr	Disk Error (low level)	Sector wasn't found
-85	clkRdErr	Clock Error (low level)	Clock couldn't be read
-86	clkWrErr	Clock Error (low level)	Verification of time write failed
-87	prWrErr	Clock Error (low level)	Verification of parameter RAM failed
-88	prInitErr	Clock Error (low level)	Validity status is other than $A8
-91	ddpSktErr	AppleTalk Manager	DDP socket error
-92	ddpLenErr	AppleTalk Manager	DDP or ALAP data length is too large
-93	noBridgeErr	AppleTalk Manager	Bridge not found
-94	lapProtErr	AppleTalk Manager	ALAP protocol attach/detach error
-95	excessCollsns	AppleTalk Manager	More than 32 RTS's or line senses without CTS
-97	portInUse	AppleTalk Manager	Port is already in use
-98	portNotCf	AppleTalk Manager	Port has not been configured for connection
-100	noScrapErr	Scrap Manager	Uninitialized desk scrap
-102	noTypeErr	Scrap Manager	No data of requested type
-108	memFullErr	Memory Manager	Heap zone not big enough for request; also iMemFullError (Print Mgr)

(continued)

Table 2.081. Continued

Value	Name	Type	Description
-109	nilHandleErr	Memory Manager	Master pointer is NIL
-111	memWZErr	Memory Manager	Can't operate in free block
-112	memPurErr	Memory Manager	Can't purge a locked block
-117	memLockedErr	Memory Manager	Memory block is locked
-192	resNotFound	Resource Manager	Couldn't find resource
-193	resFNotFound	Resource Manager	Couldn't find resource file
-194	addResFailed	Resource Manager	Attempt to add resource failed
-196	rmvResFailed	Resource Manager	Attempt to remove resource failed
-1024	nbpBuffOvr	AppleTalk Manager	NBP buffer overflow
-1025	nbpNoConfirm	AppleTalk Manager	NBP unconfirmed name
-1026	nbpConfDiff	AppleTalk Manager	NBP confirmed name for different socket
-1027	nbpDuplicate	AppleTalk Manager	NBP existing duplicate name
-1028	nbpNotFound	AppleTalk Manager	NBP name not found
-1029	nbpNISErr	AppleTalk Manager	NBP names information socket error
-1096	reqFailed	AppleTalk Manager	ATP SndRequest failed and retry count exceeded
-1097	tooManyReqs	AppleTalk Manager	ATP too many concurrent requests
-1098	tooManySkts	AppleTalk Manager	ATP too many sockets responding
-1099	badATPSkt	AppleTalk Manager	ATP responding socket bad
-1100	badBuffNum	AppleTalk Manager	ATP sequence number bad
-1101	noRelErr	AppleTalk Manager	ATP no release received
-1102	cbNotFound	AppleTalk Manager	ATP no control block found
-1103	noSendResp	AppleTalk Manager	ATP received AddRsp before SndRsp
-1104	noDataArea	AppleTalk Manager	Too many outstanding ATP calls
-1105	reqAborted	AppleTalk Manager	Request was aborted
-3101	buf2SmallErr	AppleTalk Manager	ALAP frame or DDP datagram too large for buffer
-3102	noMPPError	AppleTalk Manager	MPP driver not found
-3103	cksumErr	AppleTalk Manager	DDP checksum bad
-3104	extractErr	AppleTalk Manager	NBP no tuple in buffer
-3105	readQErr	AppleTalk Manager	Invalid socket or protocol type
-3106	atpLenErr	AppleTalk Manager	ATP message to large
-3107	atpBadRsp	AppleTalk Manager	ATPRequest response bad
-3108	recNotFnd	AppleTalk Manager	ABRecord was not found
-3109	sktClosedErr	AppleTalk Manager	Call aborted due to closed socket

Source: Inside Macintosh, pages III-205 to III-209

2.082. Macintosh Disk Format

Logical Block *	Description	Comments
0	System Startup Info	Also known as "boot blocks"
1		
2	Volume Info	
3	Volume Bit Map	Bitmap used to record "used" blocks on disk
n		
n+1	File Contents	Actual files, includes catalog tree file
1599		

Notes:
- Accurate as of System 4.1.
- * Numbers refer to HFS 800K floppy disk (MFS uses different format).

Source: Inside Macintosh, page IV-165

See Also: 2.092 to 2.095. MFS disk format tables
2.096 to 2.104. HFS disk format tables

2.083. Microfloppy Sectors Per Track

Sectors per Track	*Tracks*	*Sectors **
12	0 to 15	0 to 191
11	16 to 31	192 to 367
10	32 to 47	368 to 527
9	48 to 63	528 to 671
8	64 to 79	672 to 799

Notes: * Applies to 400K disks; 800K disks are double-sided and have twice as many sectors.

Source: Inside Macintosh, page II-211

See Also: 2.082. Macintosh Disk Format

2.084. MacBinary File Format

Size	*Name*	*Comments*
Byte	Zero 1	Placeholder = $00
64 bytes	File Name	Pascal string format
4 bytes	File Type	4-character string (no length)
4 bytes	File Creator	4-character string (no length)
Byte	File Flags	
Byte	Zero 2	Placeholder = $00
6 bytes	Desktop Location	
Byte	Protected Flag	
Byte	Zero 3	Placeholder = $00
4 bytes	Data Fork Length	
4 bytes	Resource Fork Length	
4 bytes	Creation Date	
4 bytes	Modification Date	

Notes: Formats of data follow System and Finder standards.

Source: MacBinary source code (macbin.c)

2.085. MacPaint File Format

Size	*Name*	*Item*	*Comments*
Long word	Version	Version number	Usually 0 and ignored by other programs
304 bytes	PatArray	Array of 38 patterns	Often ignored
204 bytes	Future	For future use by MacPaint	Usually ignored
Varies	Bitmap	Bitmap in PackedBits format	720 compressed scanlines (576x720 bitmap)

Source: Technical Note 86, pages 1 through 5

See Also: 2.86. PackBits File Format
2.87. PICT File Format
2.91. TIFF File Format

2.086. PackBits File Format

Size	*Name*
Varies	Data group *

Notes: • Apple does not guarantee this format for future versions of the System.
•* Data group defined as:

Size	*Name*	*Comments*
Byte	Count Flag	High bit of Count Flag set if data is packed. Count Flag is then two's compliment count of the number following data bytes to be repeated. High bit of Count Flag not set if data is unpacked. Count Flag is then zero-based count of the number of discrete bytes following as data.
Varies	Data	If Count Flag is set, data byte is to be reproduced on unpacking Count Flag times. If Count Flag is not set, discrete data bytes follow.

Source: Technical Note 171, page 1

See Also: 2.85. MacPaint File Format

2.087. PICT File Format

Size	*Name*	*Item*	*Comments*
512 bytes	Header	Application-specific header information	
Int	picSize	Length of the picture	Obsolete; use length of fork to calculate size
Rect	picFrame	Picture frame	Used as reference when scaling picture
Varies	picData	Picture data (opcodes)	See 2.088., 2.090.

Notes: • Applies to both PICT 1 and PICT 2 files (only opcodes are different).
• Only the data fork of the file is used.

Source: Inside Macintosh, pages V-86 to V-87
PICT Detective User Manual, pages 43 to 45

See Also: 2.085. MacPaint File Format
2.088. PicComments
2.089. PICT Data Types
2.090. PICT Opcodes

2.088. PicComments

Kind	*Decimal*	*Name*	*Size* *	*Description*	*Used by*	*Additional Data*
$0	0	picLParen	Byte	Begin QuickDraw grouping		
$1	1	picRParen	Byte	End QuickDraw grouping		
$64	100	picAppComment	Varies	Application-specific comment		
$82	130	picDwgBeg	Byte	Begin MacDraw picture	MacDraw, MacPaint	
$83	131	picDwgEnd	Byte	End MacDraw picture	MacDraw, MacPaint	
$8C	140	picGrpBeg	Byte	Begin grouped objects	MacDraw	
$8D	141	picGrpEnd	Byte	End grouped objects	MacDraw	
$8E	142	picBitBeg	Byte	Begin series of bitmap bands	MacDraw, MacPaint	
$8F	143	picBitEnd	Byte	End series of bitmap bands	MacDraw, MacPaint	
$96	150	picTextBegin	7 bytes	Begin a text string	MacDraw, LaserWriter Driver	TTxtPicRec (6 bytes)
$97	151	picTextEnd	Byte	End text string	MacDraw, LaserWriter Driver	
$98	152	picStringBegin	Byte	Begin banded string	MacDraw, LaserWriter Driver	
$99	153	picStringEnd	Byte	End banded string	MacDraw, LaserWriter Driver	
$9A	154	picTextCenter	9 bytes	Center of rotation to picTextBegin	MacDraw, LaserWriter Driver	TTxtCenter (8 bytes)
$9B	155	picLineLayoutOff **	Byte	Turns off line layout	LaserWriter Driver	
$9C	156	picLineLayoutOn **	Byte	Turns on line layout	LaserWriter Driver	
$9D	157	picLineLayout	17 bytes	Specify line layout for next text call	LaserWriter Driver	
$A0	160	picPolyBegin	Byte	Following LineTo() are polygon	MacDraw	
$A1	161	picPolyEnd	Byte	End polygon definition	MacDraw	
$A2	162	picPolyByt	Byte	Next data is part of freehand curve	MacDraw	
$A3	163	picPolyIgnore	Byte	Ignore following polygon	MacDraw	
$A4	164	picPolySmooth	Word	Close, fill, frame a polygon	MacDraw, LaserWriter Driver	PolyVerb (byte)
$A5	165	picPlyClo	Byte	Close MacDraw polygon	MacDraw	
$AA	170	picArrw1	Byte	One arrow, point1 to point2	MacDraw	

(continued)

Table 2.088. Continued

Kind	Decimal	Name	Size *	Description	Used by	Additional Data
$AB	171	picArrw2	Byte	One arrow, point2 to point1	MacDraw	
$AC	172	picArrw3	Byte	Two arrows between point1 & 2	MacDraw	
$AD	173	picArrwEnd	Byte	End of arrow comment	MacDraw	
$B4	180	picDashedLine **	Byte + lines	Next lines are dashed	LaserWriter Driver	TDashedLine (varies)
$B5	181	picDashedStop **	Byte	End dashed lines	LaserWriter Driver	
$B6	182	picSetLineWidth **	5 bytes	Fraction for pen size	LaserWriter Driver	Line Width (point)
$BE	190	picPostScriptBegin **	Byte	Save QD state, begin PS	LaserWriter Driver	
$BF	191	picPostScriptEnd **	Byte	End PS, restore QD state	LaserWriter Driver	
$C0	192	picPostScriptHandle **	Varies	Remaining data is PS	LaserWriter Driver	PSData
$C1	193	picPostScriptFile ** ***	Varies	Filename for 'POST' resources	LaserWriter Driver	FileName
$C2	194	picTextIsPostScript **	Byte	QD text is PS until picPostScriptEnd	LaserWriter Driver	
$C3	195	picResourcePS ** ***	9 bytes	Send PS from 'STR ' and 'STR#' resources	LaserWriter Driver	Type/ID/Index (8 bytes)
$C4	196	picSetGrayLevel	5 bytes	Set gray level to number	LaserWriter Driver	fp number
$C8	200	picRotateBegin ****	5 bytes	Begin rotation of coordinate plane	LaserWriter Driver	TRotation (point)
$C9	201	picRotateEnd ****	Byte	End rotated plane	LaserWriter Driver	
$CA	202	picRotateCenter ****	9 bytes	Center of rotated plane	LaserWriter Driver	Center (offset, 8 bytes)
$D2	210	picFormsPrinting ****	Byte	Don't flush print buffer after page	LaserWriter Driver	
$D3	211	picEndFormsPrinting ****	Byte	End forms printing	LaserWriter Driver	
$1F2	498	picCreator	5 bytes	Application creator type		Dbl word
$1F3	499	picPICTScale	5 bytes	Scaling of image		Dbl word
$3E8	1000	picBegBitmapThin	Byte	Begin bitmap thinning	ImageWriter	
$3E9	1001	picEndBitmapThin	Byte	End bitmap thinning	ImageWriter	
$3039	12345	picLasso	Byte	Image is in scrap using lasso	MacPaint, HyperCard	

Notes:
- * Size is for opcode plus any additional data.
- ** LaserWriter driver 3.0 or later only.
- *** Not available when background printing enabled.
- **** LaserWriter driver 3.1 or later only.

Source: Technical Note 91, pages 1 to 18
PICT Detective User Manual, pages 60 to 67

See Also: 2.087. PICT File Format
2.089. PICT Data Types
2.090. PICT Opcodes

2.089. PICT Data Types

Type	Size	Comments
Version 1 opcode	Byte	
Version 2 opcode	Word	
Integer	Word	
Long integer	Dbl word	
Mode	Word	
Point	Dbl word	
0...255	Byte	
-128...127	Byte	
Rect	8 bytes	Four consecutive ints: top, left, bottom, right
Poly	10+ bytes	
Region	10+ bytes	
Fixed-point number	Dbl word	
Pattern	8 bytes	
Rowbytes	Word	Always an even quantity

Source: Inside Macintosh, page V-97

See Also: 2.087. PICT File Format
2.088. PicComments
2.090. PICT Opcodes

2.090. PICT Opcodes

*Opcode # **	*Name*	*Size ***	*Additional Data*
$00	NOP	Byte	None
$01	clipRgn	Byte + region	rgn
$02	bkPat	9 bytes	Pattern (8 bytes)
$03	txFont	3 bytes	Font (word)
$04	txFace	2 bytes	Face (byte)
$05	txMode	3 bytes	Mode (word)
$06	spExtra	5 bytes	Extra (point)
$07	pnSize	5 bytes	pnSize (point)
$08	pnMode	3 bytes	Mode (word)
$09	pnPat	9 bytes	Pattern (8 bytes)
$0A	thePat	9 bytes	Pattern (8 bytes)
$0B	ovSize	5 bytes	Point (point)
$0C	origin	5 bytes	dh (word), dv (word)
$0D	txSize	3 bytes	Size (word)
$0E	fgColor	5 bytes	Color (long)
$0F	bkColor	5 bytes	Color (long)
$10	txRatio	9 bytes	Numeral (point), denominator (point)
$11	picVersion	Word	Version # (byte)
$12	BkPixPat ***	Varies	
$13	PnPixPat ***	Varies	
$14	FillPixPat ***	Varies	
$15	PnLocHFrac ***	Word	Pen Position
$16	ChExtra ***	Word	Extra space for character
$17–$19	RESERVED	Varies	
$1A	RGBFgCol ***	Varies	
$1B	RGBBkCol ***	Varies	
$1C	HiliteMode ***	None	
$1D	HiliteColor ***	Varies	
$1E	DefHilite ***	None	
$1F	OpColor ***	Varies	
$20	line	9 bytes	pnLoc (point), newPt (point)
$21	line from	5 bytes	newPt (point)
$22	short line	7 bytes	pnLoc (point), dh (signed byte), dv (signed byte)
$23	short line from	3 bytes	dh (signed byte), dv (signed byte)
$24–$27	RESERVED		
$28	long text	6 bytes + text	txLoc (point), count (byte), text
$29	DH text	3 bytes + text	dh (byte), count (byte), text
$2A	DV text	3 bytes + text	dv (byte), count (byte), text
$2B	DHDV text	4 bytes + text	dh (byte), dv (byte), count (byte), text
$2C–$2F	RESERVED		
$30	frameRect	9 bytes	Rectangle (rect)
$31	paintRect	9 bytes	Rectangle (rect)
$32	eraseRect	9 bytes	Rectangle (rect)
$33	invertRect	9 bytes	Rectangle (rect)
$34	fillRect	9 bytes	Rectangle (rect)
$35–$37	RESERVED		
$38	frameSameRect	Byte	Rectangle (rect)
$39	paintSameRect	Byte	Rectangle (rect)

(continued)

Table 2.090. Continued

*Opcode # **	*Name*	*Size ***	*Additional Data*
$3A	eraseSameRect	Byte	Rectangle (rect)
$3B	invertSameRect	Byte	Rectangle (rect)
$3C	fillSameRect	Byte	Rectangle (rect)
$3D–$3F	RESERVED		
$40	frameRRect	9 bytes	Rectangle (rect) ****
$41	paintRRect	9 bytes	Rectangle (rect) ****
$42	eraseRRect	9 bytes	Rectangle (rect) ****
$43	invertRRect	9 bytes	Rectangle (rect) ****
$44	fillRRect	9 bytes	Rectangle (rect) ****
$45–$47	RESERVED		
$48	frameSameRRect	Byte	****
$49	paintSameRRect	Byte	****
$4A	eraseSameRRect	Byte	****
$4B	invertSameRRect	Byte	****
$4C	fillSameRRect	Byte	****
$4D–$4F	RESERVED		
$50	frameOval	9 bytes	Rectangle (rect)
$51	paintOval	9 bytes	Rectangle (rect)
$52	eraseOval	9 bytes	Rectangle (rect)
$53	invertOval	9 bytes	Rectangle (rect)
$54	fillOval	9 bytes	Rectangle (rect)
$55–$57	RESERVED		
$58	frameSameOval	Byte	
$59	paintSameOval	Byte	
$5A	eraseSameOval	Byte	
$5B	invertSameOval	Byte	
$5C	fillSameOval	Byte	
$5D–$5F	RESERVED		
$60	frameArc	13 bytes	Rectangle (rect), startAngle (word), arcAngle (word)
$61	paintArc	13 bytes	Rectangle (rect), startAngle (word), arcAngle (word)
$62	eraseArc	13 bytes	Rectangle (rect), startAngle (word), arcAngle (word)
$63	invertArc	13 bytes	Rectangle (rect), startAngle (word), arcAngle (word)
$64	fillArc	13 bytes	Rectangle (rect), startAngle (word), arcAngle (word)
$65–$67	RESERVED		
$68	frameSameArc	5 bytes	StartAngle (word), arcAngle(word)
$69	paintSameArc	5 bytes	StartAngle (word), arcAngle(word)
$6A	eraseSameArc	5 bytes	StartAngle (word), arcAngle(word)
$6B	invertSameArc	5 bytes	StartAngle (word), arcAngle(word)
$6C	fillSameArc	5 bytes	StartAngle (word), arcAngle(word)
$6D–$6F	RESERVED		
$70	framePoly	Byte + polygon	Polygon
$71	paintPoly	Byte + polygon	Polygon
$72	erasePoly	Byte + polygon	Polygon
$73	invertPoly	Byte + polygon	Polygon
$74	fillPoly	Byte + polygon	Polygon
$75–$77	RESERVED		
$78	frameSamePoly †	Byte	
$79	paintSamePoly †	Byte	

(continued)

Table 2.090. Continued

*Opcode # **	*Name*	*Size ***	*Additional Data*
$7A	eraseSamePoly †	Byte	
$7B	invertSamePoly †	Byte	
$7C	fillSamePoly †	Byte	
$7D–$7F	RESERVED		
$80	frameRgn	Byte + region	Region
$81	paintRgn	Byte + region	Region
$82	eraseRgn	Byte + region	Region
$83	invertRgn	Byte + region	Region
$84	fillRgn	Byte + region	Region
$85–$87	RESERVED		
$88	frameSameRgn †	Byte	
$89	paintSameRgn †	Byte	
$8A	eraseSameRgn †	Byte	
$8B	invertSameRgn †	Byte	
$8C	fillSameRgn †	Byte	
$8D–$8F	RESERVED		
$90	BitsRect	29 bytes + bitData	rowBytes (word), bounds (8 bytes), srcRect (rect), dstRect (rect), mode (word), bitData (unpacked bytes)
$91	BitsRgn	29 bytes + region + bitData	rowBytes (word), bounds (8 bytes), srcRect (rect), dstRect (rect), mode (word), maskRgn (rgn), bitData (unpacked bytes)
$92–$97	RESERVED		
$98	PackBitsRect	29 bytes + bitData	rowBytes (word), bounds (8 bytes), srcRect (rect), dstRect (rect), mode (word), bitData (packed bytes for each row)
$99	PackBitsRgn	29 bytes + region + bitData	rowBytes (word), bounds (8 bytes), srcRect (rect), dstRect (rect), mode (word), maskRgn (rgn), bitData (packed bytes)
$9A–$9F	RESERVED		
$A0	shortComment	3 bytes	Kind (word)
$A1	longComment	5 bytes + data	Kind (word), size (word), data
$A2–$FE	RESERVED		
$FF	EndOfPicture	Byte	
$0011	Version ***	Word + HeaderOp	$02FF = version number for PICT-2
$0012	BkPixPat	Varies	Color background pattern, see source V103–105
$0013	PnPixPat	Varies	Color pen pattern, see source V103–105
$0014	FillPixPat	Varies	Color fill pattern, see source V103–105
$0015	PnLocHFrac	4 bytes	Fractional pen position (word)
$0016	ChExtra	4 bytes	Extra for each character (word)
$001A	RGBFgCol	Varies	RGB color for foreground
$001B	RGBBkCol	Varies	RGB color for background
$001C	HiliteMode	Word	None
$001D	HilteColor	Varies	RGB color for highlight
$001E	DefHilite	Word	None
$001F	OpColor	Varies	RGB color for arithmetic modes
$0C00	HeaderOp	24 bytes	RESERVED for future Apple use

(continued)

Table 2.090. Continued

Notes:
- * PICT-2 opcodes are 2 bytes in length, HO bits zero padded, and aligned on even address boundaries.
- ** Size is for opcode plus any additional data.
- *** PICT-2 only.
- **** Uses setting of ovSize point (opcode $0B).
- † Not implemented.

Source:
Technical Note 21, pages 1 to 6
Inside Macintosh, pages V-93 to V-105
PICT Detective User Manual, page 47

See Also:
2.087. PICT File Format
2.088. PicComments
2.089. PICT Data Types

2.091. TIFF File Format

Part	*Size*	*Field*	*Comments*
Header	Word	Byte order	$4949 = least to most; $4D4D = most to least
	Word	Version	$2A for version 42, the current version
	Dbl word	Pointer to first IFD	
Image File Directory	Word	# of directory entries	Must begin on word boundary
	12 bytes	First directory entry	
	12 bytes each	Additional directory entries	
	Dbl word	Point to next IFD	
Values	Varies		

Size	*Item*	*Comments*
Word	Tag	
Word	Type	1 = byte 2 = ASCII string 3 = unsigned short int 4 = unsigned long int 5 = two unsigned long ints (numerator, denominator)
Word	Length	Specified in terms of data type
Dbl word	Pointer to value	(If value is 4 bytes or less, it is stored here)

Tag	*Type*	*Name*	*Comments*
$100	Short	ImageWidth	Width of image, in pixels
$101	Short	ImageLength	Length of image, in pixels (scan lines)
$102	Short	BitsPerSample	(Default = 1)

(continued)

Table 2.091. Continued

Tag	*Type*	*Name*	*Comments*
$103	Short	Compression	1 = no compression, but tightly packed (default) 2 = CCITT Group 3 1-dimensional 3 = facsimile-compatible CCITT Group 3 4 = facsimile-compatible CCITT Group 4 32771 = same as 1, except each row begins on next word boundary 32773 = PackBits compression
$106	Short	PhotometricInterpretation	0 = MinSampleValue is white, MaxSampleValue is black 1 = MinSampleValue is black, MaxSampleValue is white 2 = RGB; Min/MaxSampleValues control intensity
$107	Short	Threshholding	1 = bilevel line art scan (default) 2 = halftone or dithered scan 3 = error diffused
$108	Short	CellWidth	If threshholding = 2, this is width of dithering/halftoning matrix in 1-bit samples
$109	Short	CellLength	If threshholding = 2, this is length of dithering/halftoning matrix in 1-bit samples
$110	ASCII	Model	Zero-terminated string of scanner model number
$111	Short/long	StripOffsets	For each strip, the byte offset of that strip
$112	Short	Orientation	1 = first row at top, first column at left (default) 2 = first row at top, first column at right 3 = first row at bottom, first column at right 4 = first row at bottom, first column at left 5 = first row at left, first column at top 6 = first row at right, first column at top 7 = first row at right, first column at bottom 8 = first row at left, first column at bottom
$115	Short	SamplesPerPixel	1= monochrome, 3 = RGB color (default = 1)
$116	Short/long	RowsPerStrip	(Default = 2**32 -1, or infinity, one long strip)
$117	Long	StripByteCounts	For each strip, the number of bytes in that strip
$118	Short	MinSampleValue	(Default = 0)
$119	Short	MaxSampleValue	(Default is 2**(BitsPerSample) -1)
$120	Long	FreeOffsets	Byte offset to free blocks in file
$121	Long	FreeByteCounts	Number of bytes in a free block
$122	Short	GrayResponseUnit	1 = tenths of a unit 2 = hundredths of a unit (default) 3 = thousandths of a unit 4 = ten-thousandths of a unit 5 = hundred-thousandths of a unit
$123	Short	GrayResponseCurve	Lookup table for GrayResponseUnit
$124	Long	Group3Options	Bit 0, 1 = 2-dimensional coding, 0 = 1-dimensional (default) Bit 2, 1 = fill bits added before EOL codes to end on byte boundary
$125	Long	Group4Options	Bit 1, 1 = uncompressed, 0 = compressed (default)
$128	Short	ResolutionUnit	1 = no absolute unit of measurement 2 = inch (default) 3 = centimeter
$129	Short	PageNumber	(Note: two numbers specified: first is page number, second is total number of pages)
$10A	Short	FillOrder	1 = most significant bits filled first 2 = least significant bits filled first

(continued)

Table 2.091. Continued

Tag	*Type*	*Name*	*Comments*
$10D	ASCII	DocumentName	Zero-terminated string
$10E	ASCII	ImageDescription	Zero-terminated string
$10F	ASCII	Make	Zero-terminated string of scanner manufacturer
$11A	Rational	XResolution	Pixels per ResolutionUnit in X direction
$11B	Rational	YResolution	Pixels per ResolutionUnit in Y direction
$11C	Short	PlaneConfiguration	1 = samples stored contiguously; single image plane 2 = samples stored in separate planes
$11D	ASCII	PageName	Zero-terminated string
$11E	Rational	XPosition	X offset to left side of image, in inches
$11F	Rational	YPosition	Y offset to top of image, in inches
$12C	Short	ColorResponseUnit	1 = tenths of a unit 2 = hundredths of a unit (default) 3 = thousandths of a unit 4 = ten-thousandths of a unit 5 = hundred-thousandths of a unit
$12D	Short	ColorResponseCurve	Lookup table for ColorResponseUnit
$FF	Short	SubfileType	1 = full resolution image data (requires ImageWidth, ImageLength, StripOffset fields) 2 = reduced resolution image data (requires ImageWidth, ImageLength, StripOffset fields) 3 = single page of multi-page image (see PageNumber tag)

Notes:
- Tags with a value of $8000 or higher are reserved for user-defined information.
- Entries for Image File Directories must be sorted in ascending order by value of the tag.

Source: Tag Image File Format, version 4.0 specification (available from Aldus Corp., Seattle, Washington)

See Also: 2.087. PICT File Format

2.092. MFS Disk Format

*Logical Block **	*Description*	*Comments*
0	System startup information	Also known as "boot blocks"
1		Set to 0 if not a startup disk
2	Volume information	Uses only first 512 bytes; second 512 bytes are start of volume bitmap
3	Volume bitmap	Bitmap used to record "used" blocks on disk
4	File directory	
n		
n+1	File contents	Actual files
799		

Notes:
- Accurate as of System 4.1.
- Assumes 1K allocation blocks, the default.
- * Numbers refer to MFS 400K floppy disk (HFS uses different format).

Source: Inside Macintosh, page II-120

See Also: 2.082. Macintosh Disk Format
2.093. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.093. MFS Volume Information

Size	*Item*	*Comments*
Word	drSigWord	Signature word, always $D2D7
Long word	drCrDate	Creation date and time (of disk)
Long word	drLsBkUp	Last backup date and time (of disk)
Word	drAtrb	Volume attributes*
Word	drNmFls	Number of files in directory
Word	drDirSt	Starting block of directory
Word	drBlLen	Length of directory, in blocks
Word	drNmAlBlks	Length of volume, in allocation blocks
Long word	drAlBlkSiz	Size of allocation blocks (normally 1K)
Long word	drClpSiz	Number bytes to allocate each time function called
Word	drAlBlSt	Starting allocation block in allocation map
Long word	drNxtFNum	Next unused file number
Word	drFreeBks	Number unused allocation blocks
Varies	drVN	Pascal string format volume name

Notes:
- Accurate as of System 4.1.
- * Bit 7 set = volume locked by hardware; bit 15 set if volume locked by software.

Source: Inside Macintosh, page II-121

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.094. MFS Allocation Map

*Entry **	*Use*
0	Block is not being used
1	Block is last block of the current file
2 to 4095 **	Number of the next block in the current file

Notes: • Accurate as of System 4.1.
•* Each entry is a 12-bit number.
•** 4095 ($FFF) is also used to indicate file directory blocks in allocation map.
Note: The first block of a file is labeled "2" because of special meanings for values of 0 and 1.

Source: Inside Macintosh, page II-122

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.095. MFS File Directory Format

Size	*Item*	*Comments*
Byte	flFlags	File Flags (bit 7 = 1 if entry is used; bit 0 = 1 if file is locked)
Byte	flTyp	File version number
16 bytes	flUsrWds	Used by Finder
Long word	flFlNum	File number
Word	flStBlk	Pointer to first allocation block of data fork for file (0 if no data fork)
Long word	flLgLen	Logical end-of-file of data fork for file
Long word	flPyLen	Physical end-of-file of data fork for file
Word	flRStBlk	Pointer to first allocation block of resource fork for file (0 if no resource fork)
Long word	flRLgLen	Logical end-of-file of resource fork for file
Long word	flRPyLen	Physical end-of-file of resource fork for file
Long word	flCrDat	Creation date and time (in seconds since midnight, 1 Jan 1904)
Long word	flMdDat	Last Modification date and time (in seconds since midnight, 1 Jan 1904)
Varies	flNam	Pascal string file name

Notes: Accurate as of System 4.1.

Source: Inside Macintosh, page II-123

See Also: 2.082. Macintosh Disk Format
2.092. to 2.094. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.096. HFS Disk Format

*Logical Block **	*Description*	*Comments*
0	System startup information	Also known as "boot blocks"
1		Set to 0 if not a startup disk
2	Volume information	Uses only first 512 bytes; second 512 bytes are start of master directory block
3	Block Map	Bitmap used to record "used" blocks on disk
4	File directory	
n		
n+1	File contents	Actual files
799		

Notes: • Accurate as of System 4.2.
• Assumes 1K allocation blocks, the default.
•* Numbers refer to HFS 400K floppy disk (MFS uses different format).

Source: Inside Macintosh, page IV-161

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.097. to 2.104. HFS disk format tables

2.097. HFS Volume Information

Size	*Item*	*Comments*
Word	drSigWord	Signature word, always $4244
Long word	drCrDate	Creation date and time (of disk)
Long word	drLsMod	Last modification date and time (of disk)
Word	drAtrb	Volume attributes*
Word	drNmFls	Number of files in directory
Word	drVBMSt	Starting block of volume bitmap
Word	drAllocPtr	Used internally
Word	drNmAlBlks	Length of volume, in allocation blocks
Long word	drAlBlkSiz	Size of allocation blocks (normally 1K)
Long word	drClpSiz	Number bytes to allocate each time function called
Word	drAlBlSt	Starting allocation block in allocation map
Long word	drNxtCNID	Next unused file number
Word	drFreeBks	Number unused allocation blocks
Byte	drVN	Length of volume name
drVN bytes	drVN+1	Volume name
Long word	drVolBkUp	Date and time of last backup
Word	drVSeqNum	Used internally
Long word	drWrCnt	Volume write count
Long word	drXTClpSiz	Clump size of extents tree file
Long word	drCTClpSiz	Clump size of catalog tree file
Word	drNmRtDirs	Number of directories in root
Long word	drFilCnt	Number of files in volume
Long word	drDirCnt	Number of directories in volume
32 bytes	drFndrInfo	Used internally by Finder
Word	drVCSize	Used internally
Word	drVCBMSize	Used internally
Word	drCtlCSize	Used internally
Long word	drXTFLSize	Length of extents tree (LEOF and PEOF)
12 bytes	drXTExtRec	Extent record for extents tree
Long word	drCTFLSize	Length of catalog tree (LEOF and PEOF)
12 bytes	drCTExtRec	First extent record for catalog tree

Notes:
- Accurate as of System 4.2.
- * Bit 7 set = volume locked by hardware; bit 15 set if volume locked by software.

Source: Inside Macintosh, IV-166

See Also:
2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.098. HFS Boot Blocks Layout

Offset	*Size*	*Item*	*Comments*
$00	Word	System startup information ID	
$02	Long word	Entry point of boot code	
$06	Word	System startup version number	$44 = blocks contain executable code
$08	Word	Used internally	Should be 0
$0A	Bytes	System resource code file name	In Pascal string format
$1A	Bytes	System shell name	In Pascal string format
$2A	Bytes	Debugger name	In Pascal string format
$3A	Bytes	Debugger name	In Pascal string format
$4A	Bytes	System startup screen name	In Pascal string format
$5A	Bytes	First program to run name	In Pascal string format
$6A	Bytes	Scrap file name	In Pascal string format
$7A	Word	Number of file control blocks	
$7C	Word	Number of events in event queue	
$7E	Long word	System heap size for 128K system	
$82	Long word	RESERVED	
$86	Long word	System heap size for 512K system	

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, pages V-351 to V-352

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.099. B*-Tree Node Record Layout

B*-Tree Node Descriptor

Size	*Item*	*Comments*
Long word	ndFLink	Previous node (leaf node only)
Long word	ndBLink	Next node (leaf node only)
Byte	ndType	$FF = leaf node, $00 = index node
Byte	ndLevel	
Word	ndNRecs	Number of B*-Tree node records
Varies	Record 0	First node record (see below)
Varies	Record n	Last node record (see below)
Varies	Free space	
	Offset to free space	
	Offset to record n	
	Offset to record 0	

B*-Tree Node Record

Byte	Key length	
Up to 255 bytes	Key	
Varies	Data or pointer	

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, pages IV-168 to IV-170

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.100. Extents Key Record Layout

Extent Descriptor

Size	*Item*	*Comments*
Word	ExtNmbr	Number of extent's first allocation block
Word	NmbrBlocks	Number of allocation blocks in extent

Extents Key

Byte	xkrKeyLen	Length of key, in bytes
Byte	xkrFkType	$00 = data fork, $FF = resource fork
Long word	xkrFNum	File number
Word	xkrFABN	Allocation block number within the file

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, page IV-171

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.101. Catalog Key Record Layout

Size	*Item*	*Comments*
Byte	ckrKeyLen	Length of key, in bytes
Byte	ckrResrv1	RESERVED, used internally
Long word	ckrParID	Parent ID
Varies	ckrCName	Name of file or directory

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, page IV-171

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.102. Directory Record Layout

Size	*Item*	*Comments*
Byte	cdrType	Must be 1 for directory record
Byte	cdrResrv2	RESERVED, used internally
Word	dirFlags	Directory flags
Word	dirVal	Directory valence
Long word	dirDirID	Directory ID
Long word	dirCrDat	Creation date and time
Long word	dirMdDat	Last modification date and time
Long word	dirBkDat	Last backup date and time
16 bytes	dirUsrInfo	Finder information
16 bytes	dirFndrInfo	Additional Finder information
16 bytes	dirResrv	RESERVED, used internally

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, page IV-173

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.103. File Record Layout

Size	*Item*	*Comments*
Byte	cdrType	Must be 2 for file records
Byte	cdrResrv2	RESERVED, used internally
Byte	filFlags	File flags (bit 7 = 1 record used; bit 0 = 1 file locked)
Byte	filTyp	File type
16 bytes	filUsrWds	Finder information
Long word	filFlNum	File number
Word	filStBlk	First allocation block (data fork)
Long word	filLgLen	Logical end-of-file (data fork)
Long word	filPyLen	Physical end-of-file (data fork)
Word	filRStBlk	First allocation block (resource fork)
Long word	filRLgLen	Logical end-of-file (resource fork)
Long word	filRPyLen	Physical end-of-file (resource fork)
Long word	filCrDat	Creation date and time
Long word	filMdDat	Last modification date and time
Long word	filBkDat	Last backup date and time
16 bytes	filFndrInfo	Additional Finder information
Word	filClpSize	File clump size
12 bytes	filExtRec	First extent record (data fork)
12 bytes	filRExtRec	First extent record (resource fork)
Long word	filResrv	RESERVED, used internally

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, page IV-172

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.104. HFS disk format tables

2.104. Thread Record Layout

Size	*Item*	*Comments*
Byte	cdrType	Must be 3 for thread records
Byte	cdrResrv2	RESERVED, used internally
8 bytes	thdResrv	RESERVED, used internally
Long word	thdParID	Associated directory's parent ID
Varies	thdCName	Associated directory's name

Notes: Accurate as of System 4.2.

Source: Inside Macintosh, page IV-173

See Also: 2.082. Macintosh Disk Format
2.092. to 2.095. MFS disk format tables
2.096. to 2.103. HFS disk format tables

2.105. Installer Script Format

Size	Item
Word	Script format *
Word	Script flags
pstr	Script name
wstr	Script help
Varies	File list
Varies	Resource file list
Word	Disk blocks **

-->

Size	Item
Word	Number of files
Varies	File spec

-->

Size	Item
Word	File spec flags
Long word	File type
Long word	File creator
Long word	Creation date
Long word	Handle
Long word	FDelsize
Long word	FAddsize
pstr	File name

-->

Size	Item
Word	Number target files
Varies	Target file specs
Varies	Source file list

-->

Size	Item
Word	File spec flags
Long word	File type
Long word	File creator
Long word	Creation date
Long word	Handle
Long word	FDelsize
Long word	FAddsize
pstr	File name

-->

Size	Item
Word	Number source files
Varies	Source file specs
Varies	Resource list

-->

Size	Item
Word	File spec flags
Long word	File type
Long word	File creator
Long word	Creation date
Long word	Handle
Long word	FDelsize
Long word	FAddsize
pstr	File name

-->

Size	Item
Word	Number resources
Varies	Resource specs

-->

Size	Item
Word	Resource spec flags
Long word	Resource type
Word	Source ID
Word	Target ID
Word	CRC/version
Word	Filler 1
Long word	Filler 2
Long word	RDelSize
Long word	RAddSize
pstr	Resource name
Word	Previous CRC list

Notes: •* 0 for applications, 1 for system installations.
•** Unused, except by system installations.

Source: Technical Note 75, pages 1-9

See Also: 2.106. Installer File Spec Flags
2.107. Installer Resource Spec Flags

2.106. Installer File Spec Flags

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	*Description*	*Allowable Values*
X																delrmve	1 = remove by user instruction
	X															delinst	1 = erase, then copy to target
		X														Copy	1 = copy to target (bit 3 must be 0)
			0	0	0	0										NOT USED	Should be 0
							X									Search for file	1 = need not be in System Folder (HFS only)
								X								No old type and creator	1 = file type/creator need not match
									X							Update only	1 = copy to target if source doesn't exist (bit 3)
										X						Resource fork	1 = resource fork will be copied or deleted
											X					Data fork	1 = data fork will be copied or deleted
												X				Need not exist	1 = file need not exist (ignored if not present)
													X			Creation date	1 = creation date must match
														X		Type and creator	1 = file type/creator must match
															1	RESERVED	Always 1

Source: Technical Note 75, pages 3 to 4

See Also: 2.105. Installer Script Format

2.107. Installer Resource Spec Flags

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	*Description*	*Allowable Values*
X																delrmve	1 = remove by user instruction
	X															delinst	1 = erase, then copy to target
		X														Copy	1 = copy to target (bit 3 must be 0)
			0	0	0	0	0									NOT USED	Should be 0
								X								No old name	1 = file name need not match (bit 1 must be 1)
									X							Update only	1 = don't copy to target if source doesn't exist (bit 3)
										0						NOT USED	Should be 0
											X					Ignore protection	1 = remove even if target file is protected
												X				Need not exist	1 = file need not exist (ignored if not present)
													0			NOT USED	Should be 0
														X		Find by ID	1 = find file using ID field, otherwise use name field
															0	RESERVED	Always 0

Source: Technical Note 75, page 6

See Also: 2.105. Installer Script Format

2.108. HyperTalk Commands

Name	*Description*	*Type*	*Syntax*	*Comments*
Abs	Absolute value	Function	Abs(number)	
Add	Adds two values	Command	Add source to dest	Dest = dest + source
Annuity	Present value of an annuity	Function	Annuity(rate, periods)	Rate is a decimal fraction
Answer	Dialog box prompt with buttons	Command	Answer question [with reply1 [or reply2 [or reply3]]]	Question is 40-char max string prompt; replys are 13-char max strings
ArrowKey	Detects arrow key press	Message	ArrowKey key	Key is left, right, up, or down
Ask	Dialog box with text edit field	Command	Ask question [with proposedResponse]	Question is string prompt; proposedResponse is <255 char string
Atan	Arctangent	Function	Atan(number)	Answer is in radians
Average	Arithmetic mean of number list	Function	Average(numberList)	NumberList is two or more numbers separated by commas
Beep	Sound system beep	Command	Beep [times]	Times is a numeric value from 1 to 32767
CharToNum	ASCII number of first character	Function	CharToNum(text)	Only the first character of text is converted
Choose	Picks a tool	Command	Choose tool	Tool must be valid tool name
Click	Emulates mouseDown,Up	Command	Click at location [with key]	Key is shiftKey, optionKey, and/or commandKey
ClickLoc	Last location of mouse click	Function	ClickLoc()	Returns pair of numbers (horz, vert offset)
Close file	Close text file	Command	Close file name	Name is string containing file name and path
Close printing	Ends print job	Command	Close printing	
CommandKey	Status of command key	Function	CommandKey()	Returns down or up
Compound	Future value of amount 1	Function	Compound(rate,periods)	Rate is a decimal fraction
Convert	Formats date or time	Command	Convert source to format	Source is a date, time, or both; format is one of standard formats
Cos	Cosine	Function	Cos(radians)	
Date	Current date	Function	[Modifier] date()	Modifier is one of the standard date formats
Delete	Removes text from container	Command	Delete chunk [of card]	Chunk specifies char, word, item, or lines of container
Dial	Dials phone	Command	Dial phoneNumber [with modem [modemCommands]]	ModemCommands is a text string of commands to go to modem
DiskSpace	Disk space available	Function	DiskSpace()	Returns number of bytes available on disk with current stack
Divide	Divides two numbers	Command	Divide dest by source	Dest = dest / source
Do	Carries out command in text	Command	Do source	Source is text containing valid HyperTalk command
DoMenu	Carries out menu command	Command	DoMenu menuItem	MenuItem is a valid HyperCard menu command
Drag	Emulates mouse drag	Command	Drag from start to finish [with key]	Keys are shiftKey, optionKey, and/or commandKey
Edit script	Opens script editor	Command	Edit script of object	Object is button, field, card, background, or stack
End	End of On or Function structure	Statement	End message	Message is text containing name of handler being terminated
End if	End of if structure	Statement	End if	
End repeat	End of repeat structure	Statement	End repeat	
EnterKey	Detects Enter key press	Message	EnterKey	
Exit	Early end of On or Function structure	Statement	Exit message	Message is text containing name of handler being terminated
Exit to HyperCard	Early end of handler function	Statement	Exit to HyperCard	Control returns to HyperCard
Exit repeat	Early end of repeat structure	Statement	Exit repeat	
Exp	Exponent of number	Function	Exp(number)	
Exp1	One less than exponent	Function	Exp1(number)	
Exp2	2 raised to a power (number)	Function	Exp2(number)	
Find	Searches for text pattern	Command	Find pattern [in field fieldName] Find character pattern [in field fieldName] Find word pattern [in field fieldName] Find string pattern [in field fieldName] Find whole pattern [in field fieldName]	
Flash	Reverses video repeatedly	Command	Flash count	Count is numeric value indicating number of times to flash
FoundChunk	Position of found text	Function	FoundChunk()	
FoundField	Identifies found field	Function	FoundField()	
FoundLine	Identifies found line	Function	FoundLine()	
FoundText	Returns text found	Function	FoundText()	
Function	Begins function declaration	Statement	Function name [parameters] statementList end name	Name is name of function
FunctionKey	Detects function key press	Message	FunctionKey number	Number is the function key pressed
Get	Puts new value in It	Command	Get source	It = source
Global	Defines variables	Command	Global variableList	VariableList is names of variables separated by commas
Go	Moves to location	Command	Go [to] card [of background][of stack] Go [to] background [of stack] Go [to] stack	
Help	Moves to first card of help stack	Command	Help	Stack must be named Help

(Continued)

Table 2.108. Continued

Name	*Description*	*Type*	*Syntax*	*Comments*
Hide	Removes item from display	Command	Hide source	Source is a button, field, or window (not card window)
Hide picture	Removes picture or background	Command	Hide card picture Hide background picture Hide picture of aCard Hide picture of aBackground	Version 1.2 only Version 1.2 only Version 1.2 only Version 1.2 only
If	Conditional execution structure	Statement	If condition then statement [else statement] If condition then statementList [else statementList] end if	
Length	Length of text characters	Function	Length(source)	Source is a text string
LineFeed	ASCII linefeed character	Constant	Linefeed	
Ln	Natural log of a number	Function	Ln(number)	
Ln1	Natural log of 1 plus number	Function	Ln1(number)	
Lock screen	Sets LockScreen property to true	Command	Lock screen	Version 1.2 only
Log2	Base-2 log of numeric value	Function	Log2(number)	
Max	Maximum value in list	Function	Max(numberList)	NumberList is numbers separated by commas
Min	Minimum value in list	Function	Min(numberList)	NumberList is numbers separated by commas
Mouse	Status of mouse button	Function	Mouse()	Returns down or up
MouseClick	Status of mouse since handler began	Function	MouseClick()	Returns true or false
MouseH	Horizontal position of mouse	Function	MouseH()	In pixels from left edge of card
MouseLoc	Position of mouse	Function	MouseLoc()	In pixels from left, and top of card
MouseV	Vertical position of mouse	Function	MouseV()	In pixels from top of card
Multiply	Multiplies two numbers	Command	Multiply dest by source	Dest = dest * source
Next repeat	Skips remainder of repeat loop	Statement	Next repeat	
Number	Counts objects	Function	Number of([domain] object) Number of(cards[of aBkgnd]) Number of(pieces in source)	Object is buttons, fields, backgrounds Pieces are characters, words, items, or lines
NumToChar	Converts number to character	Function	NumToChar(number)	Number is ASCII code of a character
Offset	Locates first occurrence of pattern	Function	Offset(pattern,source)	Pattern and source are text
On	Message handler	Statement	On message [parameters] [statementList] end message	
Open	Opens an application	Command	Open [document with] application	
Open file	Readies text file for use	Command	Open file name	Name must include path if different than current
Open printing	Begins print job	Command	Open printing [with dialog]	
OptionKey	Status of optionKey	Function	OptionKey()	Returns up or down
Param	Gets parameter value	Function	Param(number)	Number is a numeric value specifying which parameter to get
ParamCount	Counts parameters	Function	ParamCount()	Returns numeric value of number of parameters
Params	Gets all parameters	Function	Params	Parameters returned as a list
Pass	Passes message up message hierarchy, exits message handler	Statement	Pass message	Message is a message handler name
Play	Plays sound	Command	Play voice [[tempo speed] notes] Play stop	
Pop card	Retrieves long name of card	Command	Pop card [preposition destination]	Card must have been pushed first
Print	Begins print job for another app	Command	Print document with application	
Print card	Begins print job for card(s)	Command	Print oneCard Print count cards	 Count is numeric value of number of cards to print
Push card	Saves name of current card	Command	Push [recent] card	Recent is card previous to current card
Put	Places value in container	Command	Put source Put [source] preposition destination	
Random	Calculates random number	Function	Random(number)	Number is maximum value (from 1 to 32768)
Read	Import text from file	Command	Read from file name until character Read from file name for count	
Repeat	Repeats statements	Statement	Repeat [for] count [times] statementList end repeat	
Repeat while	Repeats statements while condition	Statement	Repeat while [condition] statementList end repeat	Condition is a boolean value (true or false)
Repeat until	Repeats statements until condition	Statement	Repeat until [condition] statementList end repeat	Condition is a boolean value (true or false)
Repeat forever	Repeats statements unconditionally	Statement	Repeat [forever] statementList end repeat	
Repeat with	Repeats statements	Statement	Repeat with variable = first [down] to last statementList end repeat	
Reset paint	Sets paint properties to defaults	Command	Reset paint	
Result	Text of most recent handler or command	Function	Result()	
Return	Sets value of a function	Statement	Return value	

(Continued)

Table 2.108. Continued

Name	*Description*	*Type*	*Syntax*	*Comments*
ReturnKey	Detects Return key press	Message	ReturnKey	
Round	Rounds a number	Function	Round(number)	
ScreenRect	Global coordinates of screen	Function	ScreenRect()	Returns left, top, right, bottom
Seconds	Seconds since midnight, January 1, 1904	Function	Seconds()	
Select	Selects an object	Command	Select button Select field Select [preposition] chunk of field Select [preposition] chunk of message Select empty	
SelectedChunk	Identifies position of current text selection	Function	SelectedChunk()	
SelectedField	Identifies field of current text selection	Function	SelectedField()	
SelectedLine	Identifies line of current text selection	Function	SelectedLine()	
SelectedText	Identifies currently selected text	Function	SelectedText()	
Send	Sends message to object	Command	Send message to object	Object is button, field, card, background, stack, HyperCard
Set	Changes property state	Command	Set [the] property [of object] to state	
ShiftKey	Status of Shift key	Function	ShiftKey()	Returns up or down
Show	Displays object	Command	Show source [at location]	Source is menu bar, window, button, or field
Show cards	Displays sequence of cards	Command	Show count cards	Count is a numeric value of number of cards to show
Show picture	Displays card or background picture	Command	Show card picture Show background picture Show picture of aCard Show picture of aBackground	
Sin	Sine	Function	Sin(radians)	
Sort	Arranges cards	Command	Sort [direction] [type] by criterion	Type is text, numeric, dateTime, international
Sound	Gets name of sound being played	Function	Sound()	Returns done if sound has ended
Sqrt	Square root of number	Function	Sqrt(number)	
Subtract	Subtracts one value from another	Command	Subtract source from destination	Dest = dest - source
TabKey	Detects pressing Tab key	Message	abKey	
Tan	Tangent of a number	Function	Tan(radians)	
Target	Identifies object that received message	Function	Target	
Ticks	Ticks since Mac was started	Function	Ticks()	Tick = 1/60th of a second
Time	Current time	Function	[Modifier] time()	Modifier is the time, the short time, the abbreviated time, the long time
Tool	Current selected tool	Function	Tool()	
Trunc	Truncate fractional portion of number	Function	Trunc(number)	
Type	Emulates typing at keyboard	Command	Type source [with keys]	Keys are shiftKey, optionKey, and/or commandKey
Unlock screen	Unlocks display screen	Command	Unlock screen [with effect [speed] [to image]]	
Value	Value of expression	Function	Value(source)	
Version	Version of HyperCard	Function	Version()	
Visual	Sets next effect	Command	Visual [effect] name [speed] [to image]	Name is one of many, like scroll up, scroll down, etc.
Wait	Suspends HyperCard	Command	Wait [for] duration [units] Wait until condition Wait while condition	
Write	Writes to text file	Command	Write source to file name	

Source: HyperTalk Script Language Guide, pages 60 to 172

See Also: 2.109. HyperTalk Properties
2.110. HyperTalk Constants

2.109. HyperTalk Properties

Property	*Property of*	*Values*
AutoHilite	Buttons	True, false (default)
AutoTab	Fields	True, false (default)
BlindTyping	Global	True, false (default set by user preferences card)
Bottom	Windows, buttons, fields	Offset to bottom edge of object from top of card, in pixels
BottomRight	Windows, buttons, fields	Offset to bottom right corner of object from top of card, in pixels
Brush	Painting	Value of 1 to 32, correponds to numbered brush shapes (default = 8)
CantDelete *	Stacks, backgrounds, cards	True, false (default)
CantModify *	Stacks	True, false (default)
Centered	Painting	True, false (default set by Draw Centered command in Options menu)
Cursor	Global	Watch, iBeam, none, hand, plus, busy, arrow, or cross
DragSpeed	Global	Value of 1 (slow) to 32767 (fast), or 0 (fastest and default)
EditBkgnd	Global	True, false (default set by Background command on Edit menu)
Filled	Painting	True, false (default set by Draw Filled command on Options menu)
FreeSize	Stacks	Value, in bytes, of space left on disk current stack is on
Grid	Painting	True, false (default set by Grid command on Options menu)
HeapSpace	Global	Value, in bytes, of maximum memory remaining in heapspace
Height	Windows, buttons, fields	Height, in pixels, of object
Hilite	Buttons	True, false (default)
Icon	Buttons	ID number of 'ICON' resource present in current stack, home stack, or HyperCard
ID	Backgrounds, cards, buttons, fields	ID number of the object
Language	Global	Language in which scripts are written (default = English)
Left	Windows, buttons, fields	Offset, in pixels, to left edge of object
LineSize	Painting	1, 2, 3, 4, 6, or 8 (default set by Line Size command on Options menu)
Location	Windows, buttons, fields	Horizontal, vertical offsets, in pixels, to top left corner of object
LockMessages	Global	True, false (default)
LockRecent	Global	True, false (default)
LockScreen	Global	True, false (default)
LockText	Fields	True, false (default)
Multiple	Painting	True, false (default is set by Draw Multiple command on Options menu)
MultiSpace	Painting	Value from 0 to 32,767
Name	Stacks, backgrounds, cards, buttons, fields	1 to 31-character name of object
Number	Backgrounds, cards, buttons, fields	Number of the object
NumberFormat	Global	Text pattern using 0 # and . (e.g., 0.######, the default)
Pattern	Painting	Value of 1 to 40, corresponds to paint patterns (default = 12)
PolySides	Painting	Value of 3 to 50, corresponds to number of polygonal sides (default = 4)
PowerKeys	Global	True, false (default set by User Preferences card)
Rectangle	Windows, buttons, fields	Four numbers, representing distance, in pixels, of top, left, bottom, right corners
Right	Windows, buttons, fields	Offset, in pixels, to right edge of object from left edge of card
Script	Stacks, backgrounds, cards, buttons, fields	Sets script to use for object
Scroll	Fields	Number of pixels scrolled out of view
ShowLines	Fields	True, false (default)
ShowName	Buttons	True (default), false
ShowPict *	Backgrounds, cards	True (default), false
Size	Stacks	Stack size, in bytes
StackSpace	Global	Processor stack size, in bytes
Style	Buttons	Transparent, rectangle, checkBox, opaque, roundRect, or radioButton
Style	Fields	Transparent, rectangle, opaque, scrolling, shadow
TextAlign	Painting, buttons, fields	Left, center, or right
TextArrows	Global	True, false (default)
TextFont	Painting, buttons, fields	Name of font (default = Geneva)
TextHeight	Painting, buttons, fields	Value, in pixels, of 1 to 32,767 (default = 16 or one-third higher than TextSize)
TextSize	Painting, buttons, fields	Value, in font size, of text (default = 12)
TextStyle	Painting, buttons, fields	Plain, underline, condensed, bold, outline, extended, italic, or shadow
Top	Windows, buttons, fields	Offset, in pixels, to top edge of object from edge of card
TopLeft	Windows, buttons, fields	Offset, in pixels, to top left edge of object from edge of card (two values)
UserLevel	Global	Value from 1 to 5 (default set by User Preferences card)
UserModify	Global	True, false (default)
Visible	Windows, buttons, fields	True (default), false
WideMargins	Fields	True, false (default)
Width	Windows, buttons, fields	Width, in pixels, of object

Notes: * HyperCard version 1.2 only.

Source: HyperTalk Script Language Guide, pages 173 to 213

See Also: 2.108. HyperTalk Commands
2.110. HyperTalk Constants
2.111. HyperTalk Messages

2.110. HyperTalk Constants

Constant	*Value*
Down	Key is down state
Eight	8
Empty	String of ""
False	Condition does not exist
Five	5
FormFeed	String of "\012"
Four	4
LineFeed	String of "\010"
Nine	9
One	1
Pi	3.14159265358979323846
Quote	"
Return	String of "\013"
Seven	7
Six	6
Space	" "
Tab	String of "\009"
Three	3
True	Condition exists
Two	2
Up	Key is up state
Zero	0

Source: HyperTalk Script Language Guide, pages 42, 213

See Also: 2.108. HyperTalk Commands
2.109. HyperTalk Properties
2.111. HyperTalk Messages

2.111. HyperTalk Messages

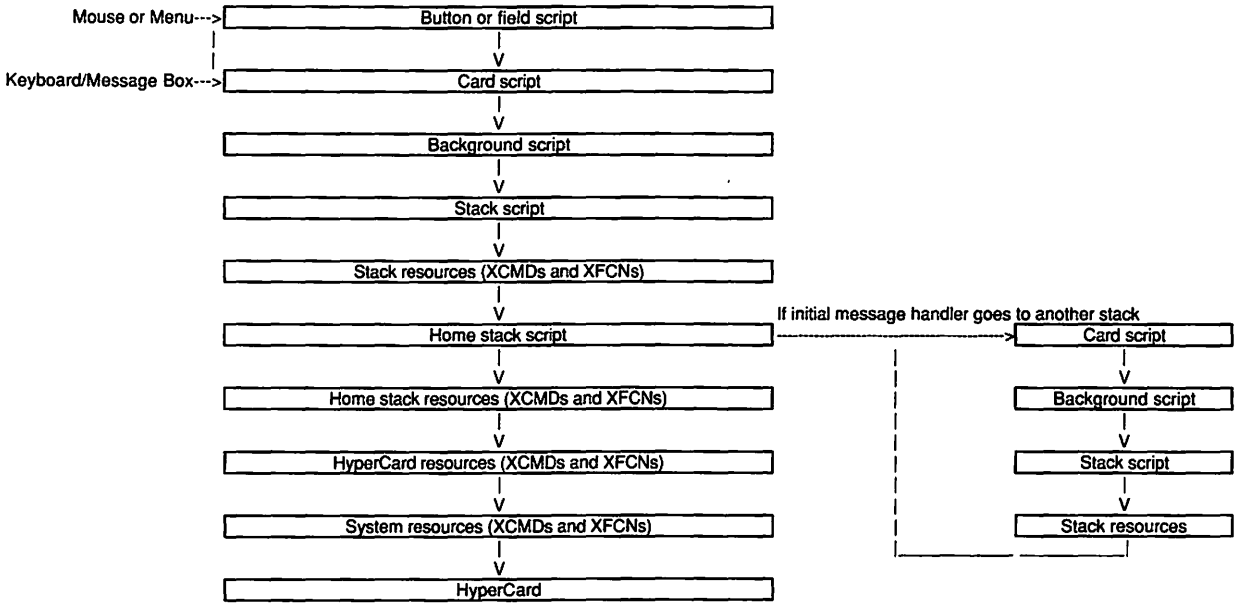
Message	*Sent When*	*Parameter*
ArrowKey	Any of the four arrow keys are pressed	Value of the key pressed
CloseBackground	Current background is left	
CloseCard	Current card is left	
CloseField	Current field is left having been changed	
CloseStack	Current stack is left	
ControlKey	Any control-key combination is pressed	Value of key pressed
DeleteBackground	Last card of background is deleted	
DeleteButton	Button is deleted	
DeleteCard	Card is deleted	
DeleteField	Field is deleted	
DeleteStack	Stack is deleted	
DoMenu	User selects a menu item	Name of menu command
EnterInField *	Enter key pressed in insertion point or selection in field	
EnterKey	Enter key is pressed (except in editing text, Ask, or Answer)	
FunctionKey	F1 through F15 are pressed (except in Ask, Answer)	Value of function key (1–15)
Help	User chooses help	
Idle	HyperCard is doing nothing	
MouseDown	Mouse button is pressed	
MouseEnter	Browse tool enters button or field rectangle (unless button down *)	
MouseLeave	Browse tool leaves button or field rectangle (unless button down *)	
MouseStillDown	Mouse button is continuously held down	
MouseUp	Mouse button is released	
MouseWithin	Browse tool is within button or field (and button not pressed)	
NewBackground	New background is created	
NewButton	New button is created	
NewCard	New card is created	
NewField	New field is created	
NewStack	New stack is created	
OpenBackground	Going to a card with different background	
OpenCard	Going to a card other than current one	
OpenField	Mouse button pressed with browse tool in unlocked field	

(continued)

Table 2.111. Continued

Message	*Sent When*	*Parameter*
OpenStack	Going to a new stack	
Quit	Quitting HyperCard	
Resume	Returning to HyperCard after quitting an application	
ReturnInField *	Return key is pressed in insertion point or selection in a field	
ReturnKey	Return key is pressed (except in editing text, Ask, or Answer)	
StartUp	HyperCard is first started or restarted	
Suspend	Open or Print command opens another application under MultiFinder	
TabKey	Tab key is pressed (except in Ask or Answer)	

Message Inheritance:



Notes: * In version 1.2 only.

Source: HyperTalk Script Language Guide, pages 11 to 32, 75 to 84

See Also: 2.108. HyperTalk Commands
2.109. HyperTalk Properties
2.110. HyperTalk Constants

2.112. MPW Commands

Command	*Description*	*Type*	*Syntax*
AddMenu	Add menu item	Miscellaneous	ADDMENU [menuName [itemName [command...]]]
Adjust	Adjust lines	Editing	ADJUST [-C count] [-L spaces] selection [window]
Alert	Display alert	Miscellaneous	ALERT [-S] [message...]
Alias	Define alternate names	Miscellaneous	ALIAS [name [word...]]
Align	Align text to left margin	Editing	ALIGN [-C count] selection [window]
Backup	Archive folder files	File management	BACKUP [option...] -FROM folder -TO folder [file...]
Beep	Generate beep sound	Miscellaneous	BEEP [note [,duration [,level]]]...
Begin...End	Grouping	Structural	BEGIN command... END
Break	Exit from For or Loop	Structural	BREAK [IF expression]
Catenate	Concatenate files	File management	CATENATE [file...]
Clear	Delete selection	Editing	CLEAR [-C count] selection [window]
Close	Close a window	Window	CLOSE [-Y | -N] [-A | windows...]
Confirm	Display confirmation dialog	Miscellaneous	CONFIRM [-T] [message...]
Continue	Continue with next iteration of For or Loop	Structural	CONTINUE [IF expression]
Copy	Copy selection to Clipboard	Editing	COPY [-C count] selection [window]
Cut	Copy selection to Clipboard and delete	Editing	CUT [-C count] selection [window]
Date	Write date and time	Miscellaneous	DATE [-A | -S] [-D | -T]
Delete	Delete files or directories	File management	DELETE [-Y | -N | -C] [-I] [-P] name...
DeleteMenu	Delete menu item	Miscellaneous	DELETEMENU [menuName [itemName]]
Directory	Set default directory	File management	DIRECTORY [-Q | directory]
Duplicate	Make copies of files or directories	File management	DUPLICATE [-Y | -N | -C] [-D | -R] [-P] name... targetName
Echo	Echo parameters	Miscellaneous	ECHO [-N] [parameters...]
Eject	Eject volumes	File management	EJECT [-M] volume...
Equal	Compare files or directories	File management	EQUAL [option...] name... targetName
Erase	Initialize volumes	File management	ERASE [-Y] [-S] volume...
Evaluate	Evaluate expression	Miscellaneous	EVALUATE [expression...]
Execute	Execute script	Miscellaneous	EXECUTE commandFile
Exists	Find file or directories	File management	EXISTS [-D | -F | -W] [-Q] name...
Exit	Exit from script	Structural	EXIT [number][IF expression]
Export	Make variables available to programs	Miscellaneous	EXPORT [-R | -S | name...]
Files	List files or directories	File management	FILES [option...] [name...]
Find	Find and select a text pattern	Editing	FIND [-C count] selection [window]
Font	Set window's font	Editing	FONT fontname fontsize [window...]
For...	Repeat sequence of commands	Structural	FOR name IN word... command... END
GetErrorText	Display text for system errors	Miscellaneous	GETERRORTEXT [-F filename] [-S filename] [-N] [-P] errnbr[,insert,...]... or GETERRORTEXT -I idnbr...
GetFileName	Display standard file dialog	Miscellaneous	GETFILENAME [-T type... | -P | -D } [-Q] [-M message] [-B buttontitle] [pathname]
GetListItem	Present file selection dialog	Miscellaneous	GETLISTITEM [option...][items...]
Help	Display summary information	Miscellaneous	HELP [-F helpfile] [command...]
If...	Conditional command execution	Structural	IF expression command... [ELSE IF expression command...]... [ELSE command...] END
Loop...End	Repeat commands	Structural	LOOP command... END
Mark	Mark and name text selection	Editing	MARK [-Y | -N] selection name [window]
Markers	List marked selections	Editing	MARKERS [-Q] [window]
Mount	Mount volumes	File management	MOUNT drive...
Move	Move files and directories	File management	MOVE [-Y | -N | -C] [-P] name... targetName

(Continued)

Table 2.112. Continued

Command	*Description*	*Type*	*Syntax*
MoveWindow	Move a window to h,v location on display	Window	MOVEWINDOW horz vert [window]
New	Open a new window	Window	NEW [name...]
Newer	Find most recent of two files	File management	NEWER [-E] [-C] [-Q] name... targetName
NewFolder	Create directory	File management	NEWFOLDER name...
Parameters	Write parameters	Miscellaneous	PARAMETERS [parameters...]
Paste	Replace selection with Clipboard contents	Editing	PASTE [-C count] selection [window]
Print	Print text files	File management	PRINT [option...] [file...]
Quit	Leave MPW	Miscellaneous	QUIT [-Y | -N | -C]
Quote	Echo parameters, quoting if needed	Miscellaneous	QUOTE [-N] [parameters...]
Rename	Give files or directories new names	File management	RENAME [-Y | -N | -C] oldname newname
Replace	Replace the selection	Editing	REPLACE [-C count] selection replacement [window]
Request	Request text in dialog	Miscellaneous	REQUEST [-D default] [message...]
Revert	Revert to last saved file	Editing	REVERT [-Y] [window...]
Save	Save files in all windows	File management	SAVE [-a | windows...]
Set	Define and write shell variables	Miscellaneous	SET [name [value]]
SetFile	Set attributes for files	File management	SETFILE [option...] file...
SetPrivilege	Set access privileges to directories	File management	SETPRIVILEGE [option...] folder...
Shift	Renumber script parameters	Miscellaneous	SHIFT [number]
ShutDown	Machine shut down or reboot	Miscellaneous	SHUTDOWN [-Y | -N | -C] [-R]
SizeWindow	Set window's dimensions	Window	SIZEWINDOW horz vert [window]
StackWindows	Arrange windows in staggered array	Window	STACKWINDOWS
Tab	Set window's tab value	Editing	TAB number [window...]
Target	Specify target window	Window	TARGET name
TileWindows	Arrange windows in tiled pattern	Window	TILEWINDOWS
Translate	Convert characters	editing	TRANSLATE [options] source [destination]
Unalias	Remove alternate names	Miscellaneous	UNALIAS [name...]
Undo	Undo last edit change	Editing	UNDO [window]
Unexport	Remove variables from program availability	Miscellaneous	UNEXPORT [-R | -S | name...]
Unmark	Remove marker from text selection	Editing	UNMARK name... window
Unmount	Unmount volumes	File management	UNMOUNT volume...
Unset	Remove shell variables	Miscellaneous	UNSET [name...]
Volumes	List mounted volumes	File management	VOLUMES [-L] [-Q] [name...]
Which	Determine file (path) shell is to execute	File management	WHICH [-A] [-P] [command]
Windows	List windows	Window	WINDOWS [-Q]
ZoomWindow	Enlarge or reduce window	Window	ZOOMWINDOW [-S] [window]

Notes: For MPW version 2.0.2.

Source: MPW: Macintosh Programmer's Workshop version 2.0.2, pages II-1 through II-207

See Also:
2.113. MPW Command Status Values
2.114. MPW Command Terminators
2.115. Special Characters in MPW Commands
2.116. Expressions in MPW Commands
2.117. Editing Selection Operations in MPW Commands
2.118. MPW Tools

2.113. MPW Command Status Values

Value	*Meaning*
-1	Not found or unexecutable
-2	Filename expansion failed or expression syntax error
-3	Bad syntax (check quotes, braces, control constructs)
-4	Missing filename or file could not be opened (in I/O redirection)
-5	Invalid expression
-6	Tool could not be started
-7	Runtime error during tool execution (probably out-of-memory error)
-8	User abort of tool from debugger
-9	User abort of tool using Command-Period

Notes: For MPW version 2.0.2.

Source: MPW: Macintosh Programmer's Workshop, pages 5-3 to 5-4

See Also: 2.112. MPW Commands
2.118. MPW Tools

2.114. MPW Command Terminators

Terminator	*Use*	*Meaning*
|	command1 | command2	Save command1 output in temp file; temp file used as input for command2
&&	command1 && command2	Executes command2 only if command1 returns a status of 0
||	command1 || command2	Executes command2 only if command1 returns a non-zero status (error)
;	command1 ; command2	Executes command2 after command1 unconditionally

Notes:
- For MPW version 2.0.2.
- All "input" and "output" references apply to standard I/O devices.

Source: MPW: Macintosh Programmer's Workshop, page 5-5

See Also: 2.112. MPW Commands
2.118. MPW Tools

2.115. Special Characters in MPW Commands

Character	*Use*	*See Also/Example*
Space	Word separator	
Tab	Word separator	
Return	Command separator	
;	Command separator	2.113. MPW Command Status Values
|	Command separator with piping	2.113. MPW Command Status Values
&&	Conditional command separator	2.113. MPW Command Status Values
||	Conditional command separator	2.113. MPW Command Status Values
(...)	Command grouping	
...	Invokes Commando	Must be typed using Option-;
#	Begins comment	#This is a comment
∂	Begins two character escape sequence	∂Return continues command on next line ∂n inserts a return character ∂t inserts a tab character ∂f inserts a formfeed character
'...'	Quotes special characters between single quotes	'&&' is literal &&, not conditional command separator
"..."	Quotes special chars between quotes, except for ∂, {, ", and '	"?" is literal ?, not character match in filename
/.../	Quotes special chars between /s, except for ∂, {, `, and /	/;/ is literal semicolon, not command separator
\...\	Quotes special chars between \s, except for ∂,{,`, and \	
{...}	Variable substitution between braces	Find "{lookfor}" "{target}" uses variables lookfor and target in Find command
`...`	Command substitution between graves	Echo the date is `date` puts output from date at EOL
?	Matches single character in a filename	Program fil? would find file, film, filo, and so on
≈	Matches any string in a filename	Program fil≈ would find fil, file, filmmaker, and so on
[...]	Character list in filename	Program fil[e,m] would find file, film, but not filo
*	Zero or more repetitions in a filename	Program fil?* would find fil, fill, filll, and so on
+	One or more repetitions in a filename	Program fil?+ would find fill, filll, and so on
« »	Specified number of repetitions in a filename	Program fil?«2» would find filll
<	Input file specification for redirection	Program <inputfile
>	Output file specification for redirection	Program >outputfile
>>	Append to output file specification for redirection	Program >>appendoutfile
≥	Diagnostic file specification for redirection	Program >outputfile ≥errorfile
≥≥	Append to diagnostic file specification for redirection	Program >outputfile ≥≥appenderrorfile

Notes: • For MPW version 2.0.2.
• All "input" and "output" references apply to standard I/O devices.

Source: MPW: Macintosh Programmer's Workshop, page 5-16

See Also: 2.112. MPW Commands
2.118. MPW Tools

2.116. Expressions in MPW Commands

Expression	*Operation*	*Precedence Level*
(Expression)	Parentheses allow expression grouping	1
- ~ ! or NOT or ¬	Unary negation Bitwise negation Logical NOT	2
* ÷ or DIV % or MOD	Multiplication Division Modulus division	3
+ -	Addition Subtraction	4
<< >>	Shift left Shift right	5
< <= or ≤ > >= or ≥	Less than Less than or equal to Greater than Greater than or equal to	6
== != or <> or ≠ =~ !~	Equal Not equal Equal pattern Not equal pattern	7
&	Bitwise AND	8
^	Bitwise XOR	9
|	Bitwise OR	10
&& or AND	Logical AND	11
|| or OR	Logical OR	12

Notes:
- For MPW version 2.0.2.
- Smaller numbers equal higher precedence.

Source: MPW: Macintosh Programmer's Workshop, page 5-23

See Also: 2.112. MPW Commands
2.118. MPW Tools

2.117. Editing Selection Operators in MPW Commands

Syntax	*Operator*	*Typed by*	*Operation*
§	§	Option-6	Current selection in target window
n		Number keys	Line number n
¡n	¡	Option-1	Line number n lines before start of current selection
!n	!	! key	Line number n lines after end of current selection
•	•	Option-8	Position before the first character of the file
∞	∞	Option-5	Position after the last character of the file
Δselection	Δ	Option-J	Position before the first character in selection
selectionΔ	Δ	Option-J	Position after the last character in selection
selection¡n	¡	Option-1	Position n characters before the first char in selection
selection!n	!	! key	Position n characters after the last char in selection
/pattern/	/	/ key	Pattern, using search forward
\pattern\	\	\ key	Pattern, using search backward
selection1:selection2	:	: key	Both selections, and everything in between
selection name			Name the selection
(selection)	()	() keys	Controls grouping of operators

(continued)

Table 2.117. Continued

Notes:
- For MPW version 2.0.2.
- Order of precedence: \ and /, (), Δ, ! and ¡, :, left-to-right evaluation.

Source: MPW: Macintosh Programmer's Workshop, page 6-4

See Also: 2.112. MPW Commands
2.118. MPW Tools

2.118. MPW Tools

Command	*Description*	*Syntax*
Asm	Assembles assembly language source files	ASM [option...][file...]
C	Compiles C source files	C [option...][file]
Canon	Copies using canonical spellings	CANON [-S][-A][-C #] dictionaryFile [inputFile...]
Compare	Compares lines of two files	COMPARE [option...] file1 [file2]
Count	Counts lines and characters in file	COUNT [-L] [-C] [file...]
CvtObj	Converts Lisa object to Macintosh object	CVTOBJ [-N namesFile][-O outputFile][-P] lisaObjFile
DeRez	Decompiles resource fork of a file	DEREZ [option...] resourceFile [resourceDescriptionFile...]
DumpCode	Disassembles object code	DUMPCODE [option...] resourceFile
DumpObj	Disassembles object code in data fork	DUMPOBJ [option...] objectFile
Entab	Replaces runs of spaces with tabs	ENTAB [option...][file...]
FileDiv	Splits file into smaller files	FILEDIV [-F][-N splitpoint] [-P] file [prefix]
Lib	Combines object files into single file	LIB [option...] objectFile...
Link	Links object files into application, etc.	LINK [option...] objectFile...
Make	Generates commands to build new version	MAKE [option...] [targetFile...]
MDSCvt	Converts MDS files to MPW syntax	MDSCVT [option...][file...]
Pascal	Compiles Pascal source files	PASCAL [option...][file...]
PasMat	Reformats Pascal source code	PASMAT [option...] [inputfile [outputfile]]
PasRef	Generates Pascal source cross reference	PASREF [option...][sourceFile...]
PerformReport	Produces performance report	PERFORMREPORT [option...]
ProcNames	Produces listing of all proc and functions in unit	PROCNAMES [option...][file...]
ResEqual	Compares resources of two files	RESEQUAL [option] file1 file2
Rez	Compiles resource text to fork of application	REZ [option...][resourceDescripFile...]
RezDet	Checks resource file for damage	REZDET [-B][-Q | -S | -D | -R | -L] resourceFile...
Search	Searches files for pattern	SEARCH [-S | -I][-R][-Q] [-F file] /pattern/ [file...]
SetVersion	Maintains version and revision numbers	SETVERSION [option...] file
TLACvt	Converts Lisa assembler to MPW source	TLACVT [option...][sourceFile...]

Notes: For MPW version 2.0.2.

Source: MPW: Macintosh Programmer's Workshop version 2.0.2, pages 11-1 through 11-207

See Also: 2.112. MPW Commands

2.119. MPW MacsBug Exceptions

Exception Number	*Description*
2	Bus error
3	Address error—data not aligned to even word boundary
4	Illegal instruction—bit pattern not recognized
5	Zero divide—attempt to divide by zero
6	CHK instruction—array index found was out of bounds
7	TRAPV instruction—overflow
9	Trace
10	Line 1010 emulator (A-trap handler used by Mac ROM toolbox)
11	Line 1111 emulator (F-trap used by coprocessors)
28	Level 4 interrupt
29	Level 5 interrupt
30	Level 6 interrupt
31	Level 7 interrupt
47	Trap $F instruction—program breakpoint

Notes: For MPW version 2.0.2.

Source: MPW: Macintosh Programmer's Workshop, page 11-4

See Also: 2.118. MPW Tools

2.120. MPW MacsBug Commands

Command	*Description*	*Syntax*	*Example*	*Comments*
?	Display help message	?	?	
A	Display/set address register	An [expr]	A0 $4F	0–7 are valid address registers
AA	Application A-trap trace	AA [trap1 [trap2 [addr1 [addr2 [D1 [D2]]]]]]	AA 0 17	
AB	A-trap break	AB [trap1 [trap2 [addr1 [addr2 [D1 [D2]]]]]]		
AH	A-trap heap zone check	AH [trap1 [trap2 [addr1 [addr2 [D1 [D2]]]]]]		
AR	A-trap record	AR [trap1 [trap2 [addr1 [addr2 [D1 [D2]]]]]]	AR 0 1000 @2AA @114	
AS	A-trap spy	AS addr1 [addr2]	AS 0 1000	
AT	A-trap trace	AT [trap1 [trap2 [addr1 [addr2 [D1 [D2]]]]]]	AT A86C A8FB	
AX	A-trap clear	AX	AX	
BA	Break in application	BA [trap1 [trap2 [addr1 [addr2 [D1 [D2]]]]]]	BA Open	
BR	Set breakpoint	BR [addr [count]]	BR $0010344C 5	Count defaults to 1 if missing
CL	Clear breakpoint	CL [addr]	CL $0010344C	
CS	Checksum	CS [addr1 [addr2]]	CS 100 200	Uses 16 bytes if one or more addrs are missing
CV	Convert expression	CV expr	CV $FF + #-32	
D	Display/set data register	Dn [expr]	D0 3A700	0–7 are valid data registers
DB	Display byte	DB [addr]	DB $0010344C	
DH	Disassemble hex	DH nmbr	DH 10	
DM	Display memory	DM [addr [nmbr]] DM [addr ['RSRC']]	DM $0010344C $1F DM $0010344C 'WIND'	No address = last address displayed 'WIND', 'IOPB', or 'TERC' only
DV	Display MacsBug version info	DV	DV	Gives version, create date/time, ap signature
DX	Debugger exchange	DX	DX	
EA	Exit to application	EA	EA	
ES	Exit to shell	ES	ES	
F	Set/display floating data register	F n [expr]	F 0 1F	
F	Find	F addr count data [mask]	F 1000 FFFF 4E70	
FC	Set/display floating control register	FC [expr]	FC	
FI	Set/display floating instruction	FI [expr]	FI	
FS	Set/display floating status register	FS [expr]	FS	
G	Go	G [addr]	G $0010344C	Starts at PC if addr is missing
GT	Go til	GT addr	GT $0010344C	Same as one-time breakpoint at addr
HC	Heap check	HC	HC	
HD	Heap dump	HD [mask]	HD 'h'	Mask is 'h', 'p', 'f', 'r', or 'XXXX' where XXXX is resource
HS	Heap scramble	HS	HS	
HT	Heap total	HT [mask]	HT 'WIND'	Mask is 'h', 'p', 'f', 'r', or 'XXXX' where XXXX is resource
HX	Heap exchange	HX [addr]	HX 1080	
ID	Instruction disassemble	ID [addr]	ID RA0	
IL	Instruction list	IL [addr [nmbr]]	IL 1008 10	
MR	Magic return	MR [offset]	MR	Uses address at top of stack if no offset
PC	Display/set PC register	PC [expr]	PC 3A700	
RB	Reboot system	RB	RB	
S	Step	S [nmbr]	S 1A	Same as T if no nmbr

(Continued)

Table 2.120. Continued

Command	*Description*	*Syntax*	*Example*	*Comments*
SB	Set byte	SB addr [expr]	SB 100 200	Byte cleared to zero if no expr supplied
SC	Stack crawl	SC	SC	
SD	Symbol dump	SD [addr]	SD RA5	
SM	Set memory	SM addr expr	SM $0010344C 'TEXT'	Text literals 1–4 chars long, otherwise width of widest operand
SR	Display/set SR register	SR [expr]	SR	
SS	Step spy	SS addr1 [addr2]	SS 100 200	Step until checksum of range changes
ST	Step til	ST addr	ST $0010344C	
SX	Symbol exchange	SX	SX	
T	Trace	T	T	Traces one instruction
TD	Display total (all registers)	TD	TD	Displays all registers, disassembles current instruction
TF	Display floating-point registers	TF	TF	Displays 6888x registers, if present
WH	Where	WH expr	WH A9F4	

Notes: For MPW version 2.0.2.

Source: MPW: Macintosh Programmer's Workshop, pages 11-1 to 11-22

See Also: 2.118. MPW Tools
2.119. MPW MacsBug Exceptions

2.121. MPW Object File Format

First Record ***Must Be First Record in File***

Size	*Description*	*Comments*
Byte	Record Type	Must be 1
Byte	Flags	Bit 0 indicates nesting (1 = nested)
Word	Version	1 for current definition

Pad Record ***Used to Byte-Align Records***

Byte	Record Type	Must be 0

Last Record ***Must Be Last Record in File***

Byte	Record Type	Must be 2
Byte	Flags	Must be 0

Comment Record ***Used for Comments***

Byte	Record Type	Must be 3
Byte	Flags	Must be 0
Word	Record Size	Number of bytes in record
Varies	Comments	Text

(continued)

Table 2.121. Continued

Dictionary Record — ***Associates Name with an ID***

Size	*Description*	*Comments*
Byte	Record Type	Must be 4
Byte	Flags	Must be 0
Word	Record Size	Number of bytes in record
Word	First ID	ID for first string
Varies	Strings	One or more strings (multiple strings associated with subsequent IDs)

Module Record — ***Associates a Module with an ID***

Byte	Record Type	Must be 5
Byte	Flags	Bit 0 = code (0) or data (1), bit 1 + 2 = 0, bit 3 = local (0) or ext (1), bit 4 = main (1) or not (0)
Word	Module ID	ID for module
Word	Segment ID/Size	ID for code modules, Size for data modules

Entry-Point Record — ***Declares Entry-Point ID***

Byte	Record Type	Must be 6
Byte	Flags	Bit 0 = code (0) or data (1), bit 1 + 2 = 0, bit 3 = local (0) or ext (1), bit 4 = main (1) or not (0)
Word	Entry ID	ID for entry point
Dbl word	Offset	Offset relative to beginning of module for entry point

Size Record — ***Declares Module Size***

Byte	Record Type	Must be 7
Byte	Flags	Bit 0 = code (0) or data (1)
Dbl word	Size	Size, in bytes

Contents Record — ***Contains Contents of Module***

Byte	Record Type	Must be 8
Byte	Flags	Bit 0 = code (0) or data (1), bit 1 + 2 = 0, bit 3 = complete (1) or partial (0), bit 4 = repeat (1)
Word	Record Size	Size, in bytes
Varies	Contents	Contents of module

Reference Record — ***Lists References to an ID***

Byte	Record Type	Must be 9
Byte	Flags	Bit 0 = code (0) or data (1), bit 1 + 2 + 5 + 6 = 0, bit 3 = long (1) or short (0), bit 4 = 32 (1) or 16 (0), bit 7 = A5 relative (1)
Word	Record Size	Size, in bytes
Word	ID	ID of module or entry point being referenced
Varies	Offsets	List of byte offsets from beginning of current code or data module

Computed-Reference Record — ***Lists Computed References***

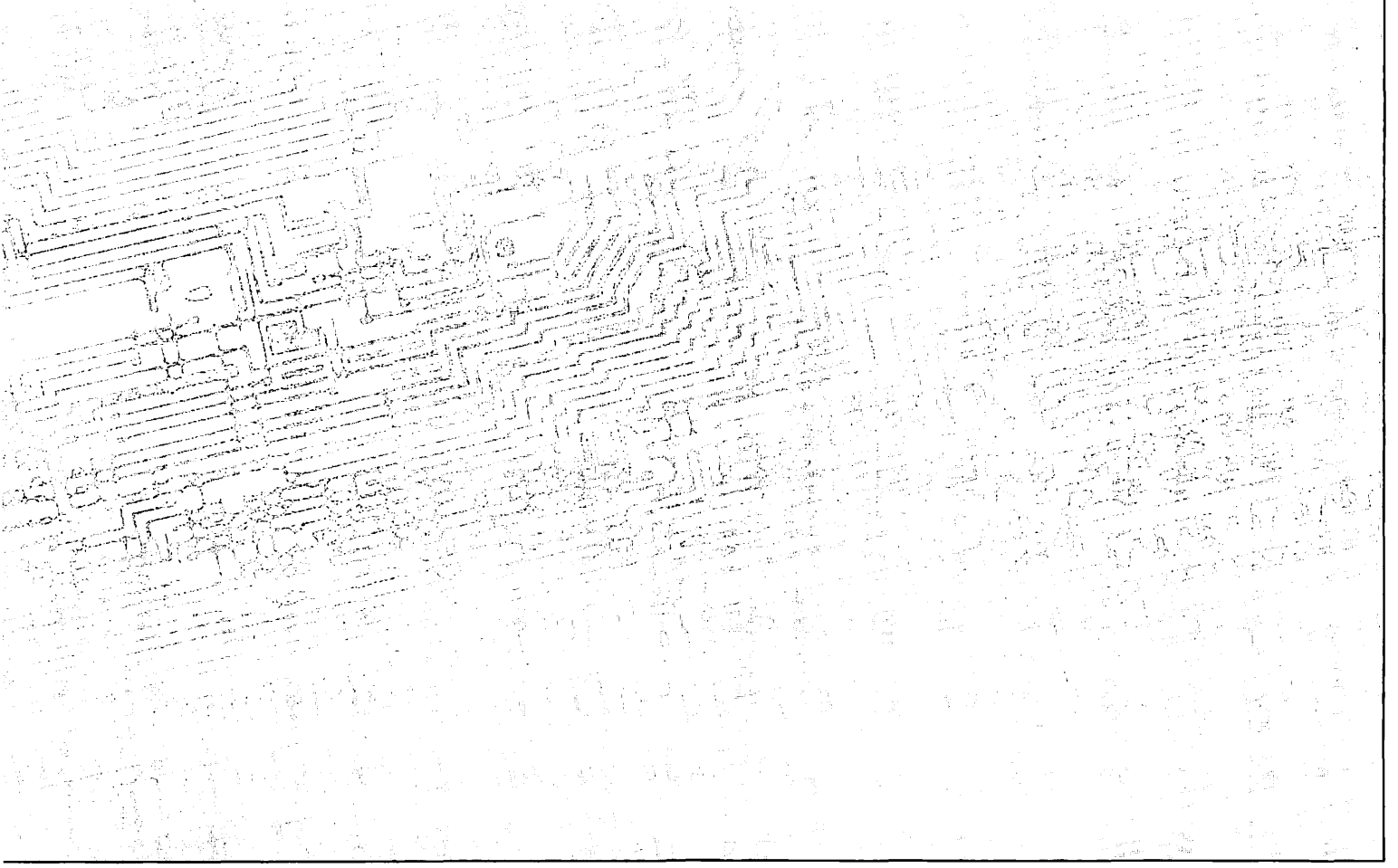
Byte	Record Type	Must be 10
Byte	Flags	Bit 0 = code (0) or data (1), bit 1 + 2 + 6 = 0, bit 3 = long (1) or short (0), bit 4 + 5 = 32 (00) or 16 (01) or 8 (10), bit 7 = difference (1)
Word	Record Size	Size, in bytes
Word	ID1	First ID of module or entry point
Word	ID2	Second ID of module or entry point
Varies	Offsets	List of byte offsets from beginning of current code or data module

Notes: For MPW version 2.0.2.

Source: MPW: Macintosh Programmer's Workshop, pages F-1 to F8

SECTION III

ROM TRAPS AND SYSTEM CALLS



3.001. ROM Versions

Size	*Type*	*Machine*	*Comments*
64K	2-256KBit	128K, 512K	Obsolete
128K	2-512KBit	512KE, Plus	Checksum: 4D 1E EE E1 original with SCSI bug
	2-512KBit		Checksum: 4D 1E EA E1 fixed boot bug
	2-512K Bit		Checksum: 4D 1F 81 72 final fix
256K	2-1MBit	SE	
256K	4-512KBit	II, IIcx	Adds color QuickDraw
256K SIMM	4-512KBit	IIx	On SIMM board

Notes: Differences in ROM versions are hidden from the user by the use of PTCH resources loaded from disk into RAM.

Source: Macintosh Family Hardware, pages 3-8, 8-12, 16-14, 18-9 to 18-12
Technical Note 139

See Also: 2.005. Standard Patches (PTCHs)

3.002. Trap Word Format

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	*Description*	*Values*
1	0	1	0													$A trap component	
				X												Trap Type Flag	(0 = Operating System Trap, 1 = Toolbox Trap)
					X	X										For OpSys Trap: flags	
					X											For Tool Trap: auto-pop bit	
						X										For Tool Trap: RESERVED	
							X									For OpSys Trap:	1 = doesn't preserve A0
							X	X	X	X	X	X	X	X	X	For Tool Trap: trap number	
								X	X	X	X	X	X	X	X	For OpSys Trap: trap number	

Source: Inside Macintosh, page I-89

See Also: 3.003. System Traps by Name
3.004. System Traps by Number
3.005. System Traps by Manager
3.006. System Traps And Routines That May Move or Purge Memory

3.003 System Traps by Name

Trap Number	*Name*	*ROM **	*Manager*
$AA94	ActivatePalette	256K	Palette Manager
$A07C	ADBOp	256K	Apple DeskTop Bus
$A07B	ADBReInit	256K	Apple DeskTop Bus
$AA3B	AddComp	256K	Color Manager
$A04E	AddDrive **	64K	Device Manager
$A87E	AddPt	64K	QuickDraw
$A9AC	AddReference	64K	Resource Manager
$A94D	AddResMenu	64K	Menu Manager
$A9AB	AddResource	64K	Resource Manager
$AA3A	AddSearch	256K	Color Manager
$A985	Alert	64K	Dialog Manager
$A010	Allocate	64K	File Manager
$A20F	AllocContig	128K	File Manager
$AA1D	AllocCursor	256K	Color QuickDraw
$A8C4	AngleFromSlope	64K	Toolbox Utilities
$AA99	AnimateEntry	256K	Palette Manager
$AA9A	AnimatePalette	256K	Palette Manager
$A933	AppendMenu	64K	Menu Manager
$A071	AttachVBL	256K	Vertical Retrace Manager
$A863	BackColor	64K	QuickDraw
$A87C	BackPat	64K	QuickDraw
$AA0B	BackPixPat	256K	Color QuickDraw
$A922	BeginUpdate	64K	Window Manager
$A85F	BitClr	64K	Toolbox Utilities
$A85A	BitNot	64K	Toolbox Utilities
$A85B	BitOr	64K	Toolbox Utilities
$A85E	BitSet	64K	Toolbox Utilities
$A85C	BitShift	64K	Toolbox Utilities
$A85D	BitTst	64K	Toolbox Utilities
$A859	BitXOr	64K	Toolbox Utilities
$A02E	BlockMove	64K	Memory Manager
$A920	BringToFront	64K	Window Manager
$A974	Button	64K	Toolbox Event Manager
$AA4F	CalcCMask	256K	Color QuickDraw
$A838	CalcMask	128K	QuickDraw
$A948	CalcMenuSize	64K	Menu Manager
$A90A	CalcVBehind	64K	Window Manager
$A909	CalcVis	64K	Window Manager
$A988	CautionAlert	64K	Dialog Manager
$A9F3	Chain	64K	Segment Loader
$A9AA	ChangedResource	64K	Resource Manager
$AA23	CharExtra	256K	Color QuickDraw
$A88D	CharWidth	64K	QuickDraw
$A945	CheckItem	64K	Menu Manager
$A911	CheckUpdate	64K	Window Manager
$A934	ClearMenuBar	64K	Menu Manager
$A90B	ClipAbove	64K	Window Manager
$A87B	ClipRect	64K	QuickDraw
$A87D	CloseCPort	256K	Color QuickDraw
$A9B7	CloseDeskAcc	64K	Desk Manager
$A982	CloseDialog	64K	Dialog Manager

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A8F4	ClosePicture	64K	QuickDraw
$A8CC	ClosePoly	64K	QuickDraw
$A87D	ClosePort	64K	QuickDraw
$A99A	CloseResFile	64K	Resource Manager
$A8DB	CloseRgn	64K	QuickDraw
$A92D	CloseWindow	64K	Window Manager
$AA33	Color2Index	256K	Color Manager
$A864	ColorBit	64K	QuickDraw
$A04C	CompactMem	64K	Memory Manager
$A004	Control	64K	Device Manager
$A8EC	CopyBits	64K	QuickDraw
$A817	CopyMask	128K	QuickDraw
$AA05	CopyPixMap	256K	Color QuickDraw
$AA09	CopyPixPat	256K	Color QuickDraw
$A8DC	CopyRgn	64K	QuickDraw
$A989	CouldAlert	64K	Dialog Manager
$A979	CouldDialog	64K	Dialog Manager
$A80D	Count1Resources	128K	Resource Manager
$A81C	Count1Types	128K	Resource Manager
$A077	CountADBs	256K	Apple DeskTop Bus
$A950	CountMItems	64K	Menu Manager
$A99C	CountResources	64K	Resource Manager
$A99E	CountTypes	64K	Resource Manager
$A008	Create	64K	File Manager
$A9B1	CreateResFile	64K	Resource Manager
$AA9F	CTab2Palette	256K	Palette Manager
$A994	CurResFile	64K	Resource Manager
$A9C7	Date2Secs	64K	OS Utilities
$A9FF	Debugger	64K	OS Utilities
$ABFF	DebugStr	64K	OS Utilities
$A03B	Delay	64K	OS Utilities
$AA4D	DelComp	256K	Color Manager
$A009	Delete	64K	File Manager
$A936	DeleteMenu	64K	Menu Manager
$AA60	DelMCEntries	256K	Menu Manager (color)
$A952	DelMenuItem	128K	Menu Manager
$AA4C	DelSearch	256K	Color Manager
$A94F	DeltaPoint	64K	Toolbox Utilities
$A96E	Dequeue	64K	OS Utilities
$A992	DetachResource	64K	Resource Manager
$A980	DialogSelect	64K	Dialog Manager
$A8E6	DiffRgn	64K	QuickDraw
$A93A	DisableItem	64K	Menu Manager
$AA63	DispMCInfo	256K	Menu Manager (color)
$AA26	DisposCCursor	256K	Color QuickDraw
$AA25	DisposCIcon	256K	Color QuickDraw
$AA24	DisposCTable	256K	Color QuickDraw
$A983	DisposDialog	64K	Dialog Manager
$A955	DisposeControl	64K	Control Manager
$A932	DisposeMenu	64K	Menu Manager
$A8D9	DisposeRgn	64K	QuickDraw

(continued)

Table 3.003. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A914	DisposeWindow	64K	Window Manager
$AA30	DisposGDevice	256K	Graphics Devices
$A023	DisposHandle	64K	Memory Manager
$AA93	DisposPalette	256K	Palette Manager
$AA04	DisposPixMap	256K	Color QuickDraw
$AA08	DisposPixPat	256K	Color QuickDraw
$A01F	DisposPtr	64K	Memory Manager
$A072	DoVBLTask	256K	Vertical Retrace Manager
$A967	DragControl	64K	Control Manager
$A905	DragGrayRgn	64K	Window Manager
$A926	DragTheRgn	64K	Window Manager
$A925	DragWindow	64K	Window Manager
$A96D	Draw1Control	128K	Control Manager
$A883	DrawChar	64K	QuickDraw
$A969	DrawControls	64K	Control Manager
$A981	DrawDialog	64K	Dialog Manager
$A904	DrawGrowIcon	64K	Window Manager
$A937	DrawMenuBar	64K	Menu Manager
$A90F	DrawNew	64K	Window Manager
$A8F6	DrawPicture	64K	QuickDraw
$A884	DrawString	64K	QuickDraw
$A885	DrawText	64K	QuickDraw
$A03D	DrvrInstall **	64K	OS Utilities
$A03E	DrvrRemove **	64K	OS Utilities
$A017	Eject	64K	File Manager
$A02B	EmptyHandle	64K	Memory Manager
$A8AE	EmptyRect	64K	QuickDraw
$A8E2	EmptyRgn	64K	QuickDraw
$A939	EnableItem	64K	Menu Manager
$A923	EndUpDate	64K	Window Manager
$A96F	Enqueue	64K	OS Utilities
$A881	EqualPt	64K	QuickDraw
$A8A6	EqualRect	64K	QuickDraw
$A8E3	EqualRgn	64K	QuickDraw
$A03C	EqualString	64K	OS Utilities
$A8C0	EraseArc	64K	QuickDraw
$A8B9	EraseOval	64K	QuickDraw
$A8C8	ErasePoly	64K	QuickDraw
$A8A3	EraseRect	64K	QuickDraw
$A8D4	EraseRgn	64K	QuickDraw
$A8B2	EraseRoundRect	64K	QuickDraw
$A98C	ErrorSound	64K	Dialog Manager
$A971	EventAvail	64K	Toolbox Event Manager
$A9F4	ExitToShell	64K	Segment Loader
$A8C2	FillArc	64K	QuickDraw
$AA11	FillCArc	256K	Color QuickDraw
$AA0F	FillCOval	256K	Color QuickDraw
$AA13	FillCPoly	256K	Color QuickDraw
$AA0E	FillCRect	256K	Color QuickDraw
$AA12	FillCRgn	256K	Color QuickDraw
$AA10	FillCRoundRect	256K	Color QuickDraw

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A8BB	FillOval	64K	QuickDraw
$A8CA	FillPoly	64K	QuickDraw
$A8A5	FillRect	64K	QuickDraw
$A8D6	FillRgn	64K	QuickDraw
$A8B4	FillRoundRect	64K	QuickDraw
$A96C	FindControl	64K	Control Manager
$A984	FindDItem	128K	Dialog Manager
$A92C	FindWindow	64K	Window Manager
$A818	FixATan2	128K	Toolbox Utilities
$A868	FixMul	64K	Toolbox Utilities
$A869	FixRatio	64K	Toolbox Utilities
$A86C	FixRound	64K	Toolbox Utilities
$A94C	FlashMenuBar	64K	Menu Manager
$A032	FlushEvents	64K	OS Event Manager
$A045	FlushFile	64K	File Manager
$A013	FlushVol	64K	File Manager
$A901	FMSwapFont	64K	Font Manager
$A835	FontMetrics	128K	Font Manager
$A862	ForeColor	64K	QuickDraw
$A8BE	FrameArc	64K	QuickDraw
$A8B7	FrameOval	64K	QuickDraw
$A8C6	FramePoly	64K	QuickDraw
$A8A1	FrameRect	64K	QuickDraw
$A8D2	FrameRgn	64K	QuickDraw
$A8B0	FrameRoundRect	64K	QuickDraw
$A98A	FreeAlert	64K	Dialog Manager
$A97A	FreeDialog	64K	Dialog Manager
$A01C	FreeMem	64K	Memory Manager
$A924	FrontWindow	64K	Window Manager
$A80E	Get1IndResource	128K	Resource Manager
$A80F	Get1IndType	128K	Resource Manager
$A820	Get1NamedResource	128K	Resource Manager
$A81F	Get1Resource	128K	Resource Manager
$A079	GetADBInfo	256K	Apple DeskTop Bus
$A9F5	GetAppParms	64K	Segment Loader
$AA44	GetAuxCtl	256K	Control Manager (color)
$AA42	GetAuxWin	256K	Window Manager (color)
$AA1A	GetBackColor	256K	Color QuickDraw
$AA1B	GetCCursor	256K	Color QuickDraw
$AA1E	GetCIcon	256K	Color QuickDraw
$A87A	GetClip	64K	QuickDraw
$AA17	GetCPixel	256K	Color QuickDraw
$A95A	GetCRefCon	64K	Control Manager
$AA18	GetCTable	256K	Color QuickDraw
$A95E	GetCTitle	64K	Control Manager
$A96A	GetCtlAction	64K	Control Manager
$A962	GetCtlMax	64K	Control Manager
$A961	GetCtlMin	64K	Control Manager
$A960	GetCtlValue	64K	Control Manager
$AA28	GetCTSeed	256K	Color Manager
$A9B9	GetCursor	64K	Toolbox Utilities

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A809	GetCVariant	256K	Control Manager (color)
$AA48	GetCWMgrPort	256K	Window Manager (color)
$A07D	GetDefaultStartup	256K	Start Manager
$AA29	GetDeviceList	256K	Graphics Devices
$A98D	GetDItem	64K	Dialog Manager
$AA9B	GetEntryColor	256K	Palette Manager
$AA9D	GetEntryUsage	256K	Palette Manager
$A011	GetEOF	64K	File Manager
$A00C	GetFileInfo	64K	File Manager
$A900	GetFNum	64K	Font Manager
$A88B	GetFontInfo	64K	QuickDraw
$A8FF	GetFontName	64K	Font Manager
$AA19	GetForeColor	256K	Color QuickDraw
$A018	GetFPos	64K	File Manager
$AA32	GetGDevice	256K	Graphics Devices
$A025	GetHandleSize	64K	Memory Manager
$A9BB	GetIcon	64K	Toolbox Utilities
$A078	GetIndADB	256K	Apple DeskTop Bus
$A99D	GetIndResource	64K	Resource Manager
$A99F	GetIndType	64K	Resource Manager
$A946	GetItem	64K	Menu Manager
$A84E	GetItemCmd	256K	Menu Manager
$A93F	GetItemIcon	64K	Menu Manager
$A943	GetItemMark	64K	Menu Manager
$A941	GetItemStyle	64K	Menu Manager
$A976	GetKeys	64K	Toolbox Event Manager
$A990	GetIText	64K	Dialog Manager
$AA2A	GetMainDevice	256K	Graphics Devices
$A836	GetMaskTable	128K	QuickDraw
$AA27	GetMaxDevice	256K	Graphics Devices
$AA64	GetMCEntry	256K	Menu Manager (color)
$AA61	GetMCInfo	256K	Menu Manager (color)
$A9BF	GetMenu	64K	Menu Manager
$A93B	GetMenuBar	64K	Menu Manager
$A949	GetMHandle	64K	Menu Manager
$A972	GetMouse	64K	Toolbox Event Manager
$A9A1	GetNamedResource	64K	Resource Manager
$A9BE	GetNewControl	64K	Control Manager
$AA46	GetNewCWindow	256K	Window Manager (color)
$A97C	GetNewDialog	64K	Dialog Manager
$A9C0	GetNewMBar	64K	Menu Manager
$AA92	GetNewPalette	256K	Palette Manager
$A9BD	GetNewWindow	64K	Window Manager
$AA2B	GetNextDevice	256K	Graphics Devices
$A970	GetNextEvent	64K	Toolbox Event Manager
$A084	GetOSDefault	256K	Start Manager
$A031	GetOSEvent	64K	OS Event Manager
$AA96	GetPalette	256K	Palette Manager
$A9B8	GetPattern	64K	Toolbox Utilities
$A89A	GetPen	64K	QuickDraw
$A898	GetPenState	64K	QuickDraw

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A9BC	GetPicture	64K	Toolbox Utilities
$A865	GetPixel	64K	QuickDraw
$AA0C	GetPixPat	256K	Color QuickDraw
$A874	GetPort	64K	QuickDraw
$A021	GetPtrSize	64K	Memory Manager
$A9A6	GetResAttrs	64K	Resource Manager
$A9F6	GetResFileAttrs	64K	Resource Manager
$A9A8	GetResInfo	64K	Resource Manager
$A9A0	GetResource	64K	Resource Manager
$A9FD	GetScrap	64K	Scrap Manager
$A9BA	GetString	64K	Toolbox Utilities
$AA37	GetSubTable	256K	Color Manager
$A146	GetTrapAddress	64K	OS Utilities
$A080	GetVideoDefault	256K	Start Manager
$A014	GetVol	64K	File Manager
$A007	GetVolInfo	64K	File Manager
$A92F	GetWindowPic	64K	Window Manager
$A910	GetWMgrPort	64K	Window Manager
$A917	GetWRefCon	64K	Window Manager
$A919	GetWTitle	64K	Window Manager
$A11A	GetZone	64K	Memory Manager
$A871	GlobalToLocal	64K	QuickDraw
$A872	GrafDevice	64K	QuickDraw
$A92B	GrowWindow	64K	Window Manager
$A9E4	HandAndHand	64K	OS Utilities
$A126	HandleZone	64K	Memory Manager
$A9E1	HandToHand	64K	OS Utilities
$A068	HClrRBit	128K	Memory Manager
$A208	HCreate	128K	File Manager (HFS)
$A209	HDelete	128K	File Manager (HFS)
$A260	HFSDispatch ***	128K	File Manager (HFS)
$A20B	HGetFileInfo	128K	File Manager (HFS)
$A069	HGetState	128K	Memory Manager
$A214	HGetVol	128K	File Manager
$A207	HGetWDInfo	128K	File Manager (HFS)
$A958	HideControl	64K	Control Manager
$A852	HideCursor	64K	QuickDraw
$A827	HideDItem	128K	Dialog Manager
$A896	HidePen	64K	QuickDraw
$A916	HideWindow	64K	Window Manager
$AA22	HiliteColor	256K	Color QuickDraw
$A95D	HiliteControl	64K	Control Manager
$A938	HiliteMenu	64K	Menu Manager
$A91C	HiliteWindow	64K	Window Manager
$A86A	HiWord	64K	Toolbox Utilities
$A029	HLock	64K	Memory Manager
$A04A	HNoPurge	64K	Memory Manager
$A9A4	HomeResFile	64K	Resource Manager
$A200	HOpen	128K	File Manager (HFS)
$A20A	HOpenRF	128K	File Manager (HFS)
$A049	HPurge	64K	Memory Manager

(continued)

Table 3.003. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A20C	HRename	128K	File Manager (HFS)
$A242	HRstFLock	128K	File Manager
$A20E	HSetFileInfo	128K	File Manager (HFS)
$A241	HSetFLock	128K	File Manager
$A067	HSetRBit	128K	Memory Manager
$A06A	HSetState	128K	Memory Manager
$A215	HSetVol	128K	File Manager
$A02A	HUnlock	64K	Memory Manager
$AA34	Index2Color	256K	Color Manager
$A9F9	InfoScrap	64K	Scrap Manager
$A9E6	InitAllPacks	64K	Package Manager
$A02C	InitApplZone	64K	Memory Manager
$AA01	InitCPort	256K	Color QuickDraw
$A850	InitCursor	64K	QuickDraw
$A97B	InitDialogs	64K	Dialog Manager
$A8FE	InitFonts	64K	Font Manager
$A06C	InitFS	128K	File Manager
$AA2E	InitGDevice	256K	Graphics Devices
$A86E	InitGraf	64K	QuickDraw
$A930	InitMenus	64K	Menu Manager
$A9E5	InitPack	64K	Package Manager
$AA90	InitPalettes	256K	Palette Manager
$A86D	InitPort	64K	QuickDraw
$A808	InitProcMenu	256K	Menu Manager
$A016	InitQueue	64K	File Manager
$A995	InitResources	64K	Resource Manager
$A03F	InitUtil	64K	OS Utilities
$A912	InitWindows	64K	Window Manager
$A019	InitZone	64K	Memory Manager
$A935	InsertMenu	64K	Menu Manager
$A951	InsertResMenu	64K	Menu Manager
$A8A9	InsetRect	64K	QuickDraw
$A8E1	InsetRgn	64K	QuickDraw
$A826	InsMenuItem	128K	Menu Manager
$A058	InsTime	128K	Time Manager
$A07F	InternalWait	256K	Start Manager
$A928	InvalRect	64K	Window Manager
$A927	InvalRgn	64K	Window Manager
$A8C1	InvertArc	64K	QuickDraw
$AA35	InvertColor	256K	Color Manager
$A8BA	InvertOval	64K	QuickDraw
$A8C9	InvertPoly	64K	QuickDraw
$A8A4	InvertRect	64K	QuickDraw
$A8D5	InvertRgn	64K	QuickDraw
$A8B3	InvertRoundRect	64K	QuickDraw
$A97F	IsDialogEvent	64K	Dialog Manager
$A9C3	KeyTrans	256K	Toolbox Event Manager
$A956	KillControls	64K	Control Manager
$A006	KillIO	64K	Device Manager
$A8F5	KillPicture	64K	QuickDraw
$A8CD	KillPoly	64K	QuickDraw

(continued)

Table 3.003. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A9F2	Launch	64K	Segment Loader
$A892	Line	64K	QuickDraw
$A891	LineTo	64K	QuickDraw
$A9A2	LoadResource	64K	Resource Manager
$A9FB	LoadScrap	64K	Scrap Manager
$A9F0	LoadSeg	64K	Segment Loader
$A870	LocalToGlobal	64K	QuickDraw
$A867	LongMul	64K	Toolbox Utilities
$A86B	LoWord	64K	Toolbox Utilities
$AA39	MakeITable	256K	Color Manager
$AA0D	MakeRGBPat	256K	Color QuickDraw
$A8FC	MapPoly	64K	QuickDraw
$A8F9	MapPt	64K	QuickDraw
$A8FA	MapRect	64K	QuickDraw
$A8FB	MapRgn	64K	QuickDraw
$A063	MaxApplZone	128K	Memory Manager
$A061	MaxBlock	128K	Memory Manager
$A11D	MaxMem	64K	Memory Manager
$A821	MaxSizeRsrc	128K	Resource Manager
$A837	MeasureText	128K	QuickDraw
$AA66	MenuChoice	256K	Menu Manager
$A93E	MenuKey	64K	Menu Manager
$A93D	MenuSelect	64K	Menu Manager
$A991	ModalDialog	64K	Dialog Manager
$A036	MoreMasters	64K	Memory Manager
$A00F	MountVol	64K	File Manager
$A894	Move	64K	QuickDraw
$A959	MoveControl	64K	Control Manager
$A064	MoveHHi	128K	Memory Manager
$A877	MovePortTo	64K	QuickDraw
$A893	MoveTo	64K	QuickDraw
$A91B	MoveWindow	64K	Window Manager
$A9E0	Munger	64K	Toolbox Utilities
$AA4B	NewCDialog	256K	Dialog Manager (color)
$A954	NewControl	64K	Control Manager
$AA45	NewCWindow	256K	Window Manager (color)
$A97D	NewDialog	64K	Dialog Manager
$A066	NewEmptyHandle	128K	Memory Manager
$AA2F	NewGDevice	256K	Graphics Devices
$A122	NewHandle	64K	Memory Manager
$A931	NewMenu	64K	Menu Manager
$AA91	NewPalette	256K	Palette Manager
$AA03	NewPixMap	256K	Color QuickDraw
$AA07	NewPixPat	256K	Color QuickDraw
$A11E	NewPtr	64K	Memory Manager
$A8D8	NewRgn	64K	QuickDraw
$A906	NewString	64K	Toolbox Utilities
$A913	NewWindow	64K	Window Manager
$A987	NoteAlert	64K	Dialog Manager
$A856	ObscureCursor	64K	QuickDraw
$A035	Offline	64K	File Manager

(continued)

Table 3.003. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A8CE	OffsetPoly	64K	QuickDraw
$A8A8	OffsetRect	64K	QuickDraw
$A8E0	OffsetRgn	64K	QuickDraw
$AA21	OpColor	256K	Color QuickDraw
$A000	Open	64K	File Manager
$AA00	OpenCPort	256K	Color QuickDraw
$A9B6	OpenDeskAcc	64K	Desk Manager
$A8F3	OpenPicture	64K	QuickDraw
$A8CB	OpenPoly	64K	QuickDraw
$A86F	OpenPort	64K	QuickDraw
$A997	OpenResFile	64K	Resource Manager
$A00A	OpenRF	64K	File Manager
$A9C4	OpenRFPerm	128K	Resource Manager
$A8DA	OpenRgn	64K	QuickDraw
$A030	OSEventAvail	64K	OS Event Manager
$A9E7	Pack0	64K	Package Manager
$A9E8	Pack1	64K	Package Manager
$A82C	Pack10	128K	Package Manager
$A82D	Pack11	128K	Package Manager
$A82E	Pack12 (ColorPicker)	128K	Package Manager
$A82F	Pack13	128K	Package Manager
$A830	Pack14	128K	Package Manager
$A831	Pack15	128K	Package Manager
$A9E9	Pack2 (DiskInit)	64K	Package Manager
$A9EA	Pack3 (StdFile)	64K	Package Manager
$A9EB	Pack4 (Floating Point)	64K	Package Manager
$A9EC	Pack5 (Transcendental Functions)	64K	Package Manager
$A9ED	Pack6 (IntUtils)	64K	Package Manager
$A9EE	Pack7 (Binary-Decimal Conversion)	64K	Package Manager
$A816	Pack8	128K	Package Manager
$A82B	Pack9	128K	Package Manager
$A8CF	PackBits	64K	Toolbox Utilities
$A8BF	PaintArc	64K	QuickDraw
$A90D	PaintBehind	64K	Window Manager
$A90C	PaintOne	64K	Window Manager
$A8B8	PaintOval	64K	QuickDraw
$A8C7	PaintPoly	64K	QuickDraw
$A8A2	PaintRect	64K	QuickDraw
$A8D3	PaintRgn	64K	QuickDraw
$A8B1	PaintRoundRect	64K	QuickDraw
$AAA0	Palette2CTab	256K	Palette Manager
$A98B	ParamText	64K	Dialog Manager
$A001	PBClose	64K	File Manager
$A89C	PenMode	64K	QuickDraw
$A89E	PenNormal	64K	QuickDraw
$A89D	PenPat	64K	QuickDraw
$AA0A	PenPixPat	256K	Color QuickDraw
$A89B	PenSize	64K	QuickDraw
$A8F2	PicComment	64K	QuickDraw
$A94E	PinRect	64K	Window Manager
$AA1F	PlotCIcon	256K	Color QuickDraw

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A94B	PlotIcon	64K	Toolbox Utilities
$AA98	PmBackColor	256K	Palette Manager
$AA97	PmForeColor	256K	Palette Manager
$A80B	PopUpMenuSelect	256K	Menu Manager
$A876	PortSize	64K	QuickDraw
$A02F	PostEvent	64K	OS Event Manager
$A12F	PPostEvent	128K	OS Event Manager
$A8FD	PrGlue	256K	Printing Manager
$A05A	PrimeTime	128K	Time Manager
$AA3D	ProtectEntry	256K	Color Manager
$A8AC	Pt2Rect	64K	QuickDraw
$A8AD	PtInRect	64K	QuickDraw
$A8E8	PtInRgn	64K	QuickDraw
$A9EF	PtrAndHand	64K	OS Utilities
$A9E3	PtrToHand	64K	OS Utilities
$A9E2	PtrToXHand	64K	OS Utilities
$A048	PtrZone	64K	Memory Manager
$A148	PtrZone	64K	Memory Manager
$A8C3	PtToAngle	64K	QuickDraw
$A04D	PurgeMem	64K	Memory Manager
$A062	PurgeSpace	128K	Memory Manager
$A9CA	PutIcon	64K	Toolbox Utilities
$A9FE	PutScrap	64K	Scrap Manager
$AA40	QDError	256K	Color Manager
$A861	Random	64K	QuickDraw
$A04F	RDrvrInstall **	64K	Device Manager
$A002	Read	64K	File Manager
$A039	ReadDateTime	64K	OS Utilities
$A037	ReadParam	64K	OS Utilities
$A051	ReadXPRam	128K	OS Utilities
$AA36	RealColor	256K	Color Manager
$A902	RealFont	64K	Font Manager
$A027	ReallocHandle	64K	Memory Manager
$A128	RecoverHandle	64K	Memory Manager
$A8E9	RectInRgn	64K	QuickDraw
$A8DF	RectRgn	64K	QuickDraw
$A9A3	ReleaseResource	64K	Resource Manager
$A050	RelString	128K	OS Utilities
$A00B	Rename	64K	File Manager
$A9AF	ResError	64K	Resource Manager
$AA3E	ReserveEntry	256K	Color Manager
$A040	ResrvMem	64K	Memory Manager
$AA4A	RestoreEntries	256K	Color Manager
$AA15	RGBBackColor	256K	Color QuickDraw
$AA14	RGBForeColor	256K	Color QuickDraw
$A80C	RGetResource	256K	Resource Manager
$A9AE	RmveReference	64K	Resource Manager
$A9AD	RmveResource	64K	Resource Manager
$A059	RmvTime	128K	Time Manager
$A9C5	RsrcMapEntry	128K	Resource Manager
$A996	RsrcZoneInit	64K	Resource Manager

(continued)

Table 3.003. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A042	RstFilLock	64K	File Manager
$AA49	SaveEntries	256K	Color Manager
$A90E	SaveOld	64K	Window Manager
$A8F8	ScalePt	64K	QuickDraw
$A8B5	ScriptUtilDispatchTable	256K	Script Manager
$A833	ScrnBitMap	128K	OS Utilities
$A8EF	ScrollRect	64K	QuickDraw
$A815	SCSIDispatchTable	128K	SCSI Manager
$A9C6	Secs2Date	64K	OS Utilities
$A8AA	SectRect	64K	QuickDraw
$A8E4	SectRgn	64K	QuickDraw
$AA50	SeedCFill	256K	Color QuickDraw
$A839	SeedFill	128K	QuickDraw
$A91F	SelectWindow	64K	Window Manager
$A97E	SellText	64K	Dialog Manager
$A921	SendBehind	64K	Window Manager
$A07A	SetADBInfo	256K	Apple DeskTop Bus
$A057	SetAppBase	64K	Memory Manager
$A02D	SetApplLimit	64K	Memory Manager
$AA1C	SetCCursor	256K	Color QuickDraw
$AA3C	SetClientID	256K	Color Manager
$A879	SetClip	64K	QuickDraw
$AA16	SetCPixel	256K	Color QuickDraw
$A95B	SetCRefCon	64K	Control Manager
$A95F	SetCTitle	64K	Control Manager
$A96B	SetCtlAction	64K	Control Manager
$AA43	SetCtlColor	256K	Control Manager (color)
$A965	SetCtlMax	64K	Control Manager
$A964	SetCtlMin	64K	Control Manager
$A963	SetCtlValue	64K	Control Manager
$A851	SetCursor	64K	QuickDraw
$A03A	SetDateTime	64K	OS Utilities
$A07E	SetDefaultStartup	256K	Start Manager
$AA47	SetDeskCPat	256K	Window Manager (color)
$AA2D	SetDeviceAttribute	256K	Graphics Devices
$A98E	SetDItem	64K	Dialog Manager
$A8DD	SetEmptyRgn	64K	QuickDraw
$AA3F	SetEntries	256K	Color Manager
$AA9C	SetEntryColor	256K	Palette Manager
$AA9E	SetEntryUsage	256K	Palette Manager
$A012	SetEOF	64K	File Manager
$A00D	SetFileInfo	64K	File Manager
$A041	SetFilLock	64K	File Manager
$A043	SetFilType	64K	File Manager
$A903	SetFontLock	64K	Font Manager
$A044	SetFPos	64K	File Manager
$A814	SetFractEnable	256K	Font Manager
$A834	SetFScaleDisable	128K	Font Manager
$AA31	SetGDevice	256K	Graphics Devices
$A04B	SetGrowZone	64K	Memory Manager
$A024	SetHandleSize	64K	Memory Manager

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A947	SetItem	64K	Menu Manager
$A84F	SetItemCmd	256K	Menu Manager
$A940	SetItemIcon	64K	Menu Manager
$A944	SetItemMark	64K	Menu Manager
$A942	SetItemStyle	64K	Menu Manager
$A98F	SetIText	64K	Dialog Manager
$AA65	SetMCEntries	256K	Menu Manager (color)
$AA62	SetMCInfo	256K	Menu Manager (color)
$A93C	SetMenuBar	64K	Menu Manager
$A94A	SetMenuFlash	64K	Menu Manager
$A878	SetOrigin	64K	QuickDraw
$A083	SetOSDefault	256K	Start Manager
$AA95	SetPalette	256K	Palette Manager
$A899	SetPenState	64K	QuickDraw
$A873	SetPort	64K	QuickDraw
$A875	SetPortBits	64K	QuickDraw
$AA06	SetPortPix	256K	Color QuickDraw
$A880	SetPt	64K	QuickDraw
$A020	SetPtrSize	64K	Memory Manager
$A8A7	SetRect	64K	QuickDraw
$A8DE	SetRectRgn	64K	QuickDraw
$A9A7	SetResAttrs	64K	Resource Manager
$A9F7	SetResFileAttrs	64K	Resource Manager
$A9A9	SetResInfo	64K	Resource Manager
$A99B	SetResLoad	64K	Resource Manager
$A993	SetResPurge	64K	Resource Manager
$AA4E	SetStdCProcs	256K	Color QuickDraw
$A8EA	SetStdProcs	64K	QuickDraw
$A907	SetString	64K	Toolbox Utilities
$A047	SetTrapAddress	64K	OS Utilities
$A081	SetVideoDefault	256K	Start Manager
$A015	SetVol	64K	File Manager
$AA41	SetWinColor	256K	Window Manager (color)
$A92E	SetWindowPic	64K	Window Manager
$A918	SetWRefCon	64K	Window Manager
$A91A	SetWTitle	64K	Window Manager
$A01B	SetZone	64K	Memory Manager
$A855	ShieldCursor	64K	Toolbox Utilities
$A957	ShowControl	64K	Control Manager
$A853	ShowCursor	64K	QuickDraw
$A828	ShowDItem	128K	Dialog Manager
$A908	ShowHide	64K	Window Manager
$A897	ShowPen	64K	QuickDraw
$A915	ShowWindow	64K	Window Manager
$A895	ShutDown Dispatch ****	256K	Shutdown Manager
$A075	SIntInstall	256K	Device Manager
$A076	SIntRemove	256K	Device Manager
$A95C	SizeControl	64K	Control Manager
$A9A5	SizeResource	64K	Resource Manager
$A91D	SizeWindow	64K	Window Manager
$A8BC	SlopeFromAngle	64K	Toolbox Utilities

(continued)

Table 3.003. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A06E	SlotManager	256K	Slot Manager
$A06F	SlotVInstall	256K	Vertical Retrace Manager
$A070	SlotVRemove	256K	Vertical Retrace Manager
$A802	SndAddModifier	256K	Sound Manager
$A806	SndControl	256K	Sound Manager
$A803	SndDoCommand	256K	Sound Manager
$A804	SndDoImmediate	256K	Sound Manager
$A807	SndNewChannel	256K	Sound Manager
$A805	SndPlay	256K	Sound Manager
$A88E	SpaceExtra	64K	QuickDraw
$A065	StackSpace	128K	Memory Manager
$A005	Status	64K	Device Manager
$A8BD	StdArc	64K	QuickDraw
$A8EB	StdBits	64K	QuickDraw
$A8F1	StdComment	64K	QuickDraw
$A8EE	StdGetPic	64K	QuickDraw
$A890	StdLine	64K	QuickDraw
$ABF8	StdOpcodeProc	256K	Color Manager
$A8B6	StdOval	64K	QuickDraw
$A8C5	StdPoly	64K	QuickDraw
$A8F0	StdPutPic	64K	QuickDraw
$A8A0	StdRect	64K	QuickDraw
$A8D1	StdRgn	64K	QuickDraw
$A8AF	StdRRect	64K	QuickDraw
$A882	StdText	64K	QuickDraw
$A8ED	StdTxMeas	64K	QuickDraw
$A973	StillDown	64K	Toolbox Event Manager
$A986	StopAlert	64K	Dialog Manager
$A88C	StringWidth	64K	QuickDraw
$A055	StripAddress	256K	OS Utilities
$A866	StuffHex	64K	QuickDraw
$A87F	SubPt	64K	QuickDraw
$A05D	SwapMMUMode	256K	OS Utilities
$A9C8	SysBeep	64K	OS Utilities
$A090	SysEnvirons	256K	OS Utilities
$A9C9	SysError	64K	System Error Handler
$A9B3	SystemClick	64K	Desk Manager
$A9C2	SystemEdit	64K	Desk Manager
$A9B2	SystemEvent	64K	Desk Manager
$A9B5	SystemMenu	64K	Desk Manager
$A9B4	SystemTask	64K	Desk Manager
$A9D8	TEActivate	64K	TextEdit
$A813	TEAutoView	128K	TextEdit
$A9D0	TECalText	64K	TextEdit
$A9D4	TEClick	64K	TextEdit
$A9D5	TECopy	64K	TextEdit
$A9D6	TECut	64K	TextEdit
$A9D9	TEDeactivate	64K	TextEdit
$A9D7	TEDelete	64K	TextEdit
$A83D	TEDispatch †	256K	TextEdit
$A9CD	TEDispose	64K	TextEdit

(continued)

Table 3.003. Continued

Trap Number	Name	ROM *	Manager
$A83C	TEGetOffset	256K	TextEdit
$A9CB	TEGetText	64K	TextEdit
$A9DA	TEIdle	64K	TextEdit
$A9CC	TEInit	64K	TextEdit
$A9DE	TEInsert	64K	TextEdit
$A9DC	TEKey	64K	TextEdit
$A9D2	TENew	64K	TextEdit
$A9DB	TEPaste	64K	TextEdit
$A812	TEPinScroll	128K	TextEdit
$A9DD	TEScroll	64K	TextEdit
$A811	TESelView	128K	TextEdit
$A9DF	TESetJust	64K	TextEdit
$A9D1	TESetSelect	64K	TextEdit
$A9CF	TESetText	64K	TextEdit
$A966	TestControl	64K	Control Manager
$AA2C	TestDeviceAttribute	256K	Graphics Devices
$A83E	TEStyleNew	256K	TextEdit
$A9D3	TEUpdate	64K	TextEdit
$A9CE	TextBox	64K	TextEdit
$A888	TextFace	64K	QuickDraw
$A887	TextFont	64K	QuickDraw
$A889	TextMode	64K	QuickDraw
$A88A	TextSize	64K	QuickDraw
$A886	TextWidth	64K	QuickDraw
$A975	TickCount	64K	OS Event Manager
$A83B	TrackBox	128K	Window Manager
$A968	TrackControl	64K	Control Manager
$A91E	TrackGoAway	64K	Window Manager
$A8AB	UnionRect	64K	QuickDraw
$A8E5	UnionRgn	64K	QuickDraw
$A810	Unique1ID	128K	Resource Manager
$A9C1	UniqueID	64K	Resource Manager
$A9FA	UnloadScrap	64K	Scrap Manager
$A9F1	UnloadSeg	64K	Segment Loader
$A00E	UnmountVol	64K	File Manager
$A8D0	UnpackBits	64K	Toolbox Utilities
$A999	UpdateResFile	64K	Resource Manager
$A953	UpdtControl	128K	Control Manager
$A978	UpdtDialog	128K	Dialog Manager
$A054	UprString	64K	OS Utilities
$A998	UseResFile	64K	Resource Manager
$A92A	ValidRect	64K	Window Manager
$A929	ValidRgn	64K	Window Manager
$A033	VInstall	64K	Vertical Retrace Manager
$A034	VRemove	64K	Vertical Retrace Manager
$A977	WaitMouseUp	64K	Toolbox Event Manager
$A003	Write	64K	File Manager
$A038	WriteParam	64K	OS Utilities
$A9B0	WriteResource	64K	Resource Manager
$A052	WriteXPRam	128K	OS Utilities
$A819	XMunger	128K	Toolbox Utilities
$A8E7	XorRgn	64K	QuickDraw
$A9FC	ZeroScrap	64K	Scrap Manager
$A83A	ZoomWindow	128K	Window Manager

(continued)

Table 3.003. Continued

Notes:

•* First ROM that includes trap.

•** For internal use only; not used by applications.

•*** Dispatched based upon additional information:

0	FSControl
1	PBOpenWD
2	PBCloseWD
5	PBCatMove
6	PBDirCreate
7	PBGetWDInfo
8	PBGetFCBInfo
9	PBGetCatInfo
10	PBSetCatInfo
11	PBSetVolInfo
16	PBLockRng
17	PBUnlockRng
$30	GetVolParms
$31	GetLogInInfo
$32	GetDirAccess
$33	SetDirAccess
$34	MapID
$35	MapName
$36	CopyFile
$37	MoveRename
$38	OpenDeny
$39	OpenRFDeny

•**** Dispatched based upon additional information:

1	ShutDwnPower
2	ShutDwnStart
3	ShutDwnInstall
4	ShutDwnRemove

•† Dispatched based upon additional information:

0	TEStylPaste
1	TESetStyle
2	TEReplaceStyle
3	TEGetStyle
4	GetStylHandle
5	SetStylHandle
6	GetStylScrap
7	TEStylInsert
8	TEGetPoint
9	TEGetHeight

Source: Inside Macintosh, pages III-215 to III-226, V-603 to V-605, X-Ref-63 to X-Ref-72

See Also: 3.008. Interface Toolbox Managers
3.156. Operating System Calls
And other tables for various Toolbox and OS routines

3.004 System Traps by Number

Trap Number	*Name*	*ROM **	*Manager*
$A000	Open	64K	File Manager
$A001	PBClose	64K	File Manager
$A002	Read	64K	File Manager
$A003	Write	64K	File Manager
$A004	Control	64K	Device Manager
$A005	Status	64K	Device Manager
$A006	KillIO	64K	Device Manager
$A007	GetVolInfo	64K	File Manager
$A008	Create	64K	File Manager
$A009	Delete	64K	File Manager
$A00A	OpenRF	64K	File Manager
$A00B	Rename	64K	File Manager
$A00C	GetFileInfo	64K	File Manager
$A00D	SetFileInfo	64K	File Manager
$A00E	UnmountVol	64K	File Manager
$A00F	MountVol	64K	File Manager
$A010	Allocate	64K	File Manager
$A011	GetEOF	64K	File Manager
$A012	SetEOF	64K	File Manager
$A013	FlushVol	64K	File Manager
$A014	GetVol	64K	File Manager
$A015	SetVol	64K	File Manager
$A016	InitQueue	64K	File Manager
$A017	Eject	64K	File Manager
$A018	GetFPos	64K	File Manager
$A019	InitZone	64K	Memory Manager
$A01B	SetZone	64K	Memory Manager
$A01C	FreeMem	64K	Memory Manager
$A01F	DisposPtr	64K	Memory Manager
$A020	SetPtrSize	64K	Memory Manager
$A021	GetPtrSize	64K	Memory Manager
$A023	DisposHandle	64K	Memory Manager
$A024	SetHandleSize	64K	Memory Manager
$A025	GetHandleSize	64K	Memory Manager
$A027	ReallocHandle	64K	Memory Manager
$A029	HLock	64K	Memory Manager
$A02A	HUnlock	64K	Memory Manager
$A02B	EmptyHandle	64K	Memory Manager
$A02C	InitApplZone	64K	Memory Manager
$A02D	SetApplLimit	64K	Memory Manager
$A02E	BlockMove	64K	Memory Manager
$A02F	PostEvent	64K	Toolbox Event Manager
$A030	OSEventAvail	64K	Toolbox Event Manager
$A031	GetOSEvent	64K	OS Event Manager
$A032	FlushEvents	64K	OS Event Manager
$A033	VInstall	64K	Vertical Retrace Manager
$A034	VRemove	64K	Vertical Retrace Manager
$A035	Offline	64K	File Manager
$A036	MoreMasters	64K	Memory Manager
$A037	ReadParam	64K	OS Utilities
$A038	WriteParam	64K	OS Utilities

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A039	ReadDateTime	64K	OS Utilities
$A03A	SetDateTime	64K	OS Utilities
$A03B	Delay	64K	OS Utilities
$A03C	EqualString	64K	OS Utilities
$A03D	DrvrInstall **	64K	OS Utilities
$A03E	DrvrRemove **	64K	OS Utilities
$A03F	InitUtil	64K	OS Utilities
$A040	ResrvMem	64K	Memory Manager
$A041	SetFilLock	64K	File Manager
$A042	RstFilLock	64K	File Manager
$A043	SetFilType	64K	File Manager
$A044	SetFPos	64K	File Manager
$A045	FlushFile	64K	File Manager
$A047	SetTrapAddress	64K	OS Utilities
$A048	PtrZone	64K	Memory Manager
$A049	HPurge	64K	Memory Manager
$A04A	HNoPurge	64K	Memory Manager
$A04B	SetGrowZone	64K	Memory Manager
$A04C	CompactMem	64K	Memory Manager
$A04D	PurgeMem	64K	Memory Manager
$A04E	AddDrive **	64K	Device Manager
$A04F	RDrvrInstall **	64K	Device Manager
$A050	RelString	128K	OS Utilities
$A051	ReadXPRam	128K	OS Utilities
$A052	WriteXPRam	128K	OS Utilities
$A053	RESERVED	-	-
$A054	UprString	64K	OS Utilities
$A055	StripAddress	256K	OS Utilities
$A056	RESERVED	-	-
$A057	SetAppBase	64K	Memory Manager
$A058	InsTime	128K	Time Manager
$A059	RmvTime	128K	Time Manager
$A05A	PrimeTime	128K	Time Manager
$A05B	RESERVED	-	-
$A05C	RESERVED	-	-
$A05D	SwapMMUMode	256K	OS Utilities
$A05E to $A060	RESERVED	-	-
$A061	MaxBlock	128K	Memory Manager
$A062	PurgeSpace	128K	Memory Manager
$A063	MaxApplZone	128K	Memory Manager
$A064	MoveHHi	128K	Memory Manager
$A065	StackSpace	128K	Memory Manager
$A066	RESERVED	-	-
$A066	NewEmptyHandle	128K	Memory Manager
$A067	HSetRBit	128K	Memory Manager
$A068	HClrRBit	128K	Memory Manager
$A069	HGetState	128K	Memory Manager
$A06A	HSetState	128K	Memory Manager
$A06B	RESERVED	-	-
$A06C	InitFS	128K	File Manager
$A06E	SlotManager	256K	Slot Manager

(continued)

Table 3.004. Continued

Trap Number	Name	ROM *	Manager
$A06F	SlotVInstall	256K	Vertical Retrace Manager
$A070	SlotVRemove	256K	Vertical Retrace Manager
$A071	AttachVBL	256K	Vertical Retrace Manager
$A072	DoVBLTask	256K	Vertical Retrace Manager
$A073 to $A074	RESERVED	-	-
$A075	SIntInstall	256K	Device Manager
$A076	SIntRemove	256K	Device Manager
$A077	CountADBs	256K	Apple DeskTop Bus
$A078	GetIndADB	256K	Apple DeskTop Bus
$A079	GetADBInfo	256K	Apple DeskTop Bus
$A07A	SetADBInfo	256K	Apple DeskTop Bus
$A07B	ADBReInit	256K	Apple DeskTop Bus
$A07C	ADBOp	256K	Apple DeskTop Bus
$A07D	GetDefaultStartup	256K	Start Manager
$A07E	SetDefaultStartup	256K	Start Manager
$A07F	InternalWait	256K	Start Manager
$A080	GetVideoDefault	256K	Start Manager
$A081	SetVideoDefault	256K	Start Manager
$A082	RESERVED	-	-
$A083	SetOSDefault	256K	Start Manager
$A084	GetOSDefault	256K	Start Manager
$A085 to $A89F	RESERVED	-	-
$A090	SysEnvirons	256K	OS Utilities
$A091 to $A119	RESERVED	-	-
$A11A	GetZone	64K	Memory Manager
$A11B to $A11C	RESERVED	-	-
$A11D	MaxMem	64K	Memory Manager
$A11E	NewPtr	64K	Memory Manager
$A11F to $A121	RESERVED	-	-
$A122	NewHandle	64K	Memory Manager
$A123 to $A127	RESERVED	-	-
$A126	HandleZone	64K	Memory Manager
$A128	RecoverHandle	64K	Memory Manager
$A129 to $A12E	RESERVED	-	-
$A12F	PPostEvent	128K	OS Event Manager
$A130 to $A145	RESERVED	-	-
$A146	GetTrapAddress	64K	OS Utilities
$A147	RESERVED	-	-
$A148	PtrZone	64K	Memory Manager
$A149 to $A165	RESERVED	-	-
$A167 to $A1FF	RESERVED	-	-
$A200	HOpen	128K	File Manager (HFS)
$A201 to $A206	RESERVED	-	-
$A207	HGetWDInfo	128K	File Manager (HFS)
$A208	HCreate	128K	File Manager (HFS)
$A209	HDelete	128K	File Manager (HFS)
$A20A	HOpenRF	128K	File Manager (HFS)
$A20B	HGetFileInfo	128K	File Manager (HFS)
$A20C	HRename	128K	File Manager (HFS)
$A20D	RESERVED	-	-
$A20E	HSetFileInfo	128K	File Manager (HFS)

(continued)

Table 3.004. Continued

Trap Number	Name	ROM *	Manager
$A20F	AllocContig	128K	File Manager
$A210 to $A213	RESERVED	-	-
$A214	HGetVol	128K	File Manager
$A215	HSetVol	128K	File Manager
$A216 to $A240	RESERVED	–	-
$A241	HSetFLock	128K	File Manager
$A242	HRstFLock	128K	File Manager
$A243 to $A25F	RESERVED	-	-
$A260	HFSDispatch ***	128K	File Manager (HFS)
$A261 to $A801	RESERVED	-	-
$A802	SndAddModifier	256K	Sound Manager
$A803	SndDoCommand	256K	Sound Manager
$A804	SndDoImmediate	256K	Sound Manager
$A805	SndPlay	256K	Sound Manager
$A806	SndControl	256K	Sound Manager
$A807	SndNewChannel	256K	Sound Manager
$A808	InitProcMenu	256K	Menu Manager
$A809	GetCVariant	256K	Control Manager (color)
$A80A	RESERVED	-	-
$A80B	PopUpMenuSelect	256K	Menu Manager
$A80C	RGetResource	256K	Resource Manager
$A80D	Count1Resources	128K	Resource Manager
$A80E	Get1IndResource	128K	Resource Manager
$A80F	Get1IndType	128K	Resource Manager
$A810	Unique1ID	128K	Resource Manager
$A811	TESelView	128K	TextEdit
$A812	TEPinScroll	128K	TextEdit
$A813	TEAutoView	128K	TextEdit
$A814	SetFractEnable	256K	Font Manager
$A815	SCSIDispatchTable	128K	SCSI Manager
$A816	Pack8	128K	Package Manager
$A817	CopyMask	128K	QuickDraw
$A818	FixATan2	128K	Toolbox Utilities
$A819	XMunger	128K	Toolbox Utilities
$A81A to $A81B	RESERVED	-	-
$A81C	Count1Types	128K	Resource Manager
$A81D to $A81E	RESERVED	-	-
$A81F	Get1Resource	128K	Resource Manager
$A820	Get1NamedResource	128K	Resource Manager
$A821	MaxSizeRsrc	128K	Resource Manager
$A822 to $A825	RESERVED	-	-
$A826	InsMenuItem	128K	Menu Manager
$A827	HideDItem	128K	Dialog Manager
$A828	ShowDItem	128K	Dialog Manager
$A829 to $A82A	RESERVED	-	-
$A82B	Pack9	128K	Package Manager
$A82C	Pack10	128K	Package Manager
$A82D	Pack11	128K	Package Manager
$A82E	Pack12 (ColorPicker)	128K	Package Manager
$A82F	Pack13	128K	Package Manager
$A830	Pack14	128K	Package Manager

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A831	Pack15	128K	Package Manager
$A832	RESERVED	-	-
$A833	ScrnBitMap	128K	OS Utilities
$A834	SetFScaleDisable	128K	Font Manager
$A835	FontMetrics	128K	Font Manager
$A836	GetMaskTable	128K	QuickDraw
$A837	MeasureText	128K	QuickDraw
$A838	CalcMask	128K	QuickDraw
$A839	SeedFill	128K	QuickDraw
$A83A	ZoomWindow	128K	Window Manager
$A83B	TrackBox	128K	Window Manager
$A83C	TEGetOffset	256K	TextEdit
$A83D	TEDispatch ****	256K	TextEdit
$A83E	TEStyleNew	256K	TextEdit
$A83F to $A84D	RESERVED	-	-
$A84E	GetItemCmd	256K	Menu Manager
$A84F	SetItemCmd	256K	Menu Manager
$A850	InitCursor	64K	QuickDraw
$A851	SetCursor	64K	QuickDraw
$A852	HideCursor	64K	QuickDraw
$A853	ShowCursor	64K	QuickDraw
$A854	RESERVED	-	-
$A855	ShieldCursor	64K	Toolbox Utilities
$A856	ObscureCursor	64K	QuickDraw
$A857 to $A858	RESERVED	-	-
$A859	BitXOr	64K	Toolbox Utilities
$A85A	BitNot	64K	Toolbox Utilities
$A85B	BitOr	64K	Toolbox Utilities
$A85C	BitShift	64K	Toolbox Utilities
$A85D	BitTst	64K	Toolbox Utilities
$A85E	BitSet	64K	Toolbox Utilities
$A85F	BitClr	64K	Toolbox Utilities
$A860	RESERVED	-	-
$A861	Random	64K	QuickDraw
$A862	ForeColor	64K	QuickDraw
$A863	BackColor	64K	QuickDraw
$A864	ColorBit	64K	QuickDraw
$A865	GetPixel	64K	QuickDraw
$A866	StuffHex	64K	QuickDraw
$A867	LongMul	64K	Toolbox Utilities
$A868	FixMul	64K	Toolbox Utilities
$A869	FixRatio	64K	Toolbox Utilities
$A86A	HiWord	64K	Toolbox Utilities
$A86B	LoWord	64K	Toolbox Utilities
$A86C	FixRound	64K	Toolbox Utilities
$A86D	InitPort	64K	QuickDraw
$A86E	InitGraf	64K	QuickDraw
$A86F	OpenPort	64K	QuickDraw
$A870	LocalToGlobal	64K	QuickDraw
$A871	GlobalToLocal	64K	QuickDraw
$A872	GrafDevice	64K	QuickDraw

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A873	SetPort	64K	QuickDraw
$A874	GetPort	64K	QuickDraw
$A875	SetPortBits	64K	QuickDraw
$A876	PortSize	64K	QuickDraw
$A877	MovePortTo	64K	QuickDraw
$A878	SetOrigin	64K	QuickDraw
$A879	SetClip	64K	QuickDraw
$A87A	GetClip	64K	QuickDraw
$A87B	ClipRect	64K	QuickDraw
$A87C	BackPat	64K	QuickDraw
$A87D	CloseCPort	256K	Color QuickDraw
$A87D	ClosePort	64K	QuickDraw
$A87E	AddPt	64K	QuickDraw
$A87F	SubPt	64K	QuickDraw
$A880	SetPt	64K	QuickDraw
$A881	EqualPt	64K	QuickDraw
$A882	StdText	64K	QuickDraw
$A883	DrawChar	64K	QuickDraw
$A884	DrawString	64K	QuickDraw
$A885	DrawText	64K	QuickDraw
$A886	TextWidth	64K	QuickDraw
$A887	TextFont	64K	QuickDraw
$A888	TextFace	64K	QuickDraw
$A889	TextMode	64K	QuickDraw
$A88A	TextSize	64K	QuickDraw
$A88B	GetFontInfo	64K	QuickDraw
$A88C	StringWidth	64K	QuickDraw
$A88D	CharWidth	64K	QuickDraw
$A88E	SpaceExtra	64K	QuickDraw
$A88F	RESERVED	-	-
$A890	StdLine	64K	QuickDraw
$A891	LineTo	64K	QuickDraw
$A892	Line	64K	QuickDraw
$A893	MoveTo	64K	QuickDraw
$A894	Move	64K	QuickDraw
$A895	Shutdown Dispatch †	256K	Shutdown Manager
$A896	HidePen	64K	QuickDraw
$A897	ShowPen	64K	QuickDraw
$A898	GetPenState	64K	QuickDraw
$A899	SetPenState	64K	QuickDraw
$A89A	GetPen	64K	QuickDraw
$A89B	PenSize	64K	QuickDraw
$A89C	PenMode	64K	QuickDraw
$A89D	PenPat	64K	QuickDraw
$A89E	PenNormal	64K	QuickDraw
$A89F	RESERVED	-	-
$A8A0	StdRect	64K	QuickDraw
$A8A1	FrameRect	64K	QuickDraw
$A8A2	PaintRect	64K	QuickDraw
$A8A3	EraseRect	64K	QuickDraw
$A8A4	InvertRect	64K	QuickDraw

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A8A5	FillRect	64K	QuickDraw
$A8A6	EqualRect	64K	QuickDraw
$A8A7	SetRect	64K	QuickDraw
$A8A8	OffsetRect	64K	QuickDraw
$A8A9	InsetRect	64K	QuickDraw
$A8AA	SectRect	64K	QuickDraw
$A8AB	UnionRect	64K	QuickDraw
$A8AC	Pt2Rect	64K	QuickDraw
$A8AD	PtInRect	64K	QuickDraw
$A8AE	EmptyRect	64K	QuickDraw
$A8AF	StdRRect	64K	QuickDraw
$A8B0	FrameRoundRect	64K	QuickDraw
$A8B1	PaintRoundRect	64K	QuickDraw
$A8B2	EraseRoundRect	64K	QuickDraw
$A8B3	InvertRoundRect	64K	QuickDraw
$A8B4	FillRoundRect	64K	QuickDraw
$A8B5	ScriptUtil	256K	Script Manager
$A8B6	StdOval	64K	QuickDraw
$A8B7	FrameOval	64K	QuickDraw
$A8B8	PaintOval	64K	QuickDraw
$A8B9	EraseOval	64K	QuickDraw
$A8BA	InvertOval	64K	QuickDraw
$A8BB	FillOval	64K	QuickDraw
$A8BC	SlopeFromAngle	64K	Toolbox Utilities
$A8BD	StdArc	64K	QuickDraw
$A8BE	FrameArc	64K	QuickDraw
$A8BF	PaintArc	64K	QuickDraw
$A8C0	EraseArc	64K	QuickDraw
$A8C1	InvertArc	64K	QuickDraw
$A8C2	FillArc	64K	QuickDraw
$A8C3	PtToAngle	64K	QuickDraw
$A8C4	AngleFromSlope	64K	Toolbox Utilities
$A8C5	StdPoly	64K	QuickDraw
$A8C6	FramePoly	64K	QuickDraw
$A8C7	PaintPoly	64K	QuickDraw
$A8C8	ErasePoly	64K	QuickDraw
$A8C9	InvertPoly	64K	QuickDraw
$A8CA	FillPoly	64K	QuickDraw
$A8CB	OpenPoly	64K	QuickDraw
$A8CC	ClosePoly	64K	QuickDraw
$A8CD	KillPoly	64K	QuickDraw
$A8CE	OffsetPoly	64K	QuickDraw
$A8CF	PackBits	64K	Toolbox Utilities
$A8D0	UnpackBits	64K	Toolbox Utilities
$A8D1	StdRgn	64K	QuickDraw
$A8D2	FrameRgn	64K	QuickDraw
$A8D3	PaintRgn	64K	QuickDraw
$A8D4	EraseRgn	64K	QuickDraw
$A8D5	InvertRgn	64K	QuickDraw
$A8D6	FillRgn	64K	QuickDraw
$A8D7	RESERVED	-	-

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A8D8	NewRgn	64K	QuickDraw
$A8D9	DisposRgn	64K	QuickDraw
$A8DA	OpenRgn	64K	QuickDraw
$A8DB	CloseRgn	64K	QuickDraw
$A8DC	CopyRgn	64K	QuickDraw
$A8DD	SetEmptyRgn	64K	QuickDraw
$A8DE	SetRectRgn	64K	QuickDraw
$A8DF	RectRgn	64K	QuickDraw
$A8E0	OffsetRgn	64K	QuickDraw
$A8E1	InsetRgn	64K	QuickDraw
$A8E2	EmptyRgn	64K	QuickDraw
$A8E3	EqualRgn	64K	QuickDraw
$A8E4	SectRgn	64K	QuickDraw
$A8E5	UnionRgn	64K	QuickDraw
$A8E6	DiffRgn	64K	QuickDraw
$A8E7	XorRgn	64K	QuickDraw
$A8E8	PtInRgn	64K	QuickDraw
$A8E9	RectInRgn	64K	QuickDraw
$A8EA	SetStdProcs	64K	QuickDraw
$A8EB	StdBits	64K	QuickDraw
$A8EC	CopyBits	64K	QuickDraw
$A8ED	StdTxMeas	64K	QuickDraw
$A8EE	StdGetPic	64K	QuickDraw
$A8EF	ScrollRect	64K	QuickDraw
$A8F0	StdPutPic	64K	QuickDraw
$A8F1	StdComment	64K	QuickDraw
$A8F2	PicComment	64K	QuickDraw
$A8F3	OpenPicture	64K	QuickDraw
$A8F4	ClosePicture	64K	QuickDraw
$A8F5	KillPicture	64K	QuickDraw
$A8F6	DrawPicture	64K	QuickDraw
$A8F7	RESERVED	64K	QuickDraw
$A8F8	ScalePt	64K	QuickDraw
$A8F9	MapPt	64K	QuickDraw
$A8FA	MapRect	64K	QuickDraw
$A8FB	MapRgn	64K	QuickDraw
$A8FC	MapPoly	64K	QuickDraw
$A8FD	PrGlue	256K	Printing Manager
$A8FE	InitFonts	64K	Font Manager
$A8FF	GetFontName	64K	Font Manager
$A900	GetFNum	64K	Font Manager
$A901	FMSwapFont	64K	Font Manager
$A902	RealFont	64K	Font Manager
$A903	SetFontLock	64K	Font Manager
$A904	DrawGrowIcon	64K	Window Manager
$A905	DragGrayRgn	64K	Window Manager
$A906	NewString	64K	Toolbox Utilities
$A907	SetString	64K	Toolbox Utilities
$A908	ShowHide	64K	Window Manager
$A909	CalcVis	64K	Window Manager
$A90A	CalcVBehind	64K	Window Manager

(continued)

Table 3.004. Continued

Trap Number	Name	ROM *	Manager
$A90B	ClipAbove	64K	Window Manager
$A90C	PaintOne	64K	Window Manager
$A90D	PaintBehind	64K	Window Manager
$A90E	SaveOld	64K	Window Manager
$A90F	DrawNew	64K	Window Manager
$A910	GetWMgrPort	64K	Window Manager
$A911	CheckUpdate	64K	Window Manager
$A912	InitWindows	64K	Window Manager
$A913	NewWindow	64K	Window Manager
$A914	DisposWindow	64K	Window Manager
$A915	ShowWindow	64K	Window Manager
$A916	HideWindow	64K	Window Manager
$A917	GetWRefCon	64K	Window Manager
$A918	SetWRefCon	64K	Window Manager
$A919	GetWTitle	64K	Window Manager
$A91A	SetWTitle	64K	Window Manager
$A91B	MoveWindow	64K	Window Manager
$A91C	HiliteWindow	64K	Window Manager
$A91D	SizeWindow	64K	Window Manager
$A91E	TrackGoAway	64K	Window Manager
$A91F	SelectWindow	64K	Window Manager
$A920	BringToFront	64K	Window Manager
$A921	SendBehind	64K	Window Manager
$A922	BeginUpdate	64K	Window Manager
$A923	EndUpdate	64K	Window Manager
$A924	FrontWindow	64K	Window Manager
$A925	DragWindow	64K	Window Manager
$A926	DragTheRgn	64K	Window Manager
$A927	InvalRgn	64K	Window Manager
$A928	InvalRect	64K	Window Manager
$A929	ValidRgn	64K	Window Manager
$A92A	ValidRect	64K	Window Manager
$A92B	GrowWindow	64K	Window Manager
$A92C	FindWindow	64K	Window Manager
$A92D	CloseWindow	64K	Window Manager
$A92E	SetWindowPic	64K	Window Manager
$A92F	GetWindowPic	64K	Window Manager
$A930	InitMenus	64K	Menu Manager
$A931	NewMenu	64K	Menu Manager
$A932	DisposMenu	64K	Menu Manager
$A933	AppendMenu	64K	Menu Manager
$A934	ClearMenuBar	64K	Menu Manager
$A935	InsertMenu	64K	Menu Manager
$A936	DeleteMenu	64K	Menu Manager
$A937	DrawMenuBar	64K	Menu Manager
$A938	HiliteMenu	64K	Menu Manager
$A939	EnableItem	64K	Menu Manager
$A93A	DisableItem	64K	Menu Manager
$A93B	GetMenuBar	64K	Menu Manager
$A93C	SetMenuBar	64K	Menu Manager
$A93D	MenuSelect	64K	Menu Manager

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A93E	MenuKey	64K	Menu Manager
$A93F	GetItemIcon	64K	Menu Manager
$A940	SetItemIcon	64K	Menu Manager
$A941	GetItemStyle	64K	Menu Manager
$A942	SetItemStyle	64K	Menu Manager
$A943	GetItemMark	64K	Menu Manager
$A944	SetItemMark	64K	Menu Manager
$A945	CheckItem	64K	Menu Manager
$A946	GetItem	64K	Menu Manager
$A947	SetItem	64K	Menu Manager
$A948	CalcMenuSize	64K	Menu Manager
$A949	GetMHandle	64K	Menu Manager
$A94A	SetMenuFlash	64K	Menu Manager
$A94B	PlotIcon	64K	Toolbox Utilities
$A94C	FlashMenuBar	64K	Menu Manager
$A94D	AddResMenu	64K	Menu Manager
$A94E	PinRect	64K	Window Manager
$A94F	DeltaPoint	64K	Toolbox Utilities
$A950	CountMItems	64K	Menu Manager
$A951	InsertResMenu	64K	Menu Manager
$A952	DelMenuItem	128K	Menu Manager
$A953	UpdtControl	128K	Control Manager
$A954	NewControl	64K	Control Manager
$A955	DisposControl	64K	Control Manager
$A956	KillControls	64K	Control Manager
$A957	ShowControl	64K	Control Manager
$A958	HideControl	64K	Control Manager
$A959	MoveControl	64K	Control Manager
$A95A	GetCRefCon	64K	Control Manager
$A95B	SetCRefCon	64K	Control Manager
$A95C	SizeControl	64K	Control Manager
$A95D	HiliteControl	64K	Control Manager
$A95E	GetCTitle	64K	Control Manager
$A95F	SetCTitle	64K	Control Manager
$A960	GetCtlValue	64K	Control Manager
$A961	GetCtlMin	64K	Control Manager
$A962	GetCtlMax	64K	Control Manager
$A963	SetCtlValue	64K	Control Manager
$A964	SetCtlMin	64K	Control Manager
$A965	SetCtlMax	64K	Control Manager
$A966	TestControl	64K	Control Manager
$A967	DragControl	64K	Control Manager
$A968	TrackControl	64K	Control Manager
$A969	DrawControls	64K	Control Manager
$A96A	GetCtlAction	64K	Control Manager
$A96B	SetCtlAction	64K	Control Manager
$A96C	FindControl	64K	Control Manager
$A96D	Draw1Control	128K	Control Manager
$A96E	Dequeue	64K	OS Utilities
$A96F	Enqueue	64K	OS Utilities
$A970	GetNextEvent	64K	Toolbox Event Manager

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A971	EventAvail	64K	Toolbox Event Manager
$A972	GetMouse	64K	Toolbox Event Manager
$A973	StillDown	64K	Toolbox Event Manager
$A974	Button	64K	Toolbox Event Manager
$A975	TickCount	64K	OS Event Manager
$A976	GetKeys	64K	Toolbox Event Manager
$A977	WaitMouseUp	64K	Toolbox Event Manager
$A978	UpdtDialog	128K	Dialog Manager
$A979	CouldDialog	64K	Dialog Manager
$A97A	FreeDialog	64K	Dialog Manager
$A97B	InitDialogs	64K	Dialog Manager
$A97C	GetNewDialog	64K	Dialog Manager
$A97D	NewDialog	64K	Dialog Manager
$A97E	SelIText	64K	Dialog Manager
$A97F	IsDialogEvent	64K	Dialog Manager
$A980	DialogSelect	64K	Dialog Manager
$A981	DrawDialog	64K	Dialog Manager
$A982	CloseDialog	64K	Dialog Manager
$A983	DisposDialog	64K	Dialog Manager
$A984	FindDItem	128K	Dialog Manager
$A985	Alert	64K	Dialog Manager
$A986	StopAlert	64K	Dialog Manager
$A987	NoteAlert	64K	Dialog Manager
$A988	CautionAlert	64K	Dialog Manager
$A989	CouldAlert	64K	Dialog Manager
$A98A	FreeAlert	64K	Dialog Manager
$A98B	ParamText	64K	Dialog Manager
$A98C	ErrorSound	64K	Dialog Manager
$A98D	GetDItem	64K	Dialog Manager
$A98E	SetDItem	64K	Dialog Manager
$A98F	SetIText	64K	Dialog Manager
$A990	GetIText	64K	Dialog Manager
$A991	ModalDialog	64K	Dialog Manager
$A992	DetachResource	64K	Resource Manager
$A993	SetResPurge	64K	Resource Manager
$A994	CurResFile	64K	Resource Manager
$A995	InitResources	64K	Resource Manager
$A996	RsrcZoneInit	64K	Resource Manager
$A997	OpenResFile	64K	Resource Manager
$A998	UseResFile	64K	Resource Manager
$A999	UpdateResFile	64K	Resource Manager
$A99A	CloseResFile	64K	Resource Manager
$A99B	SetResLoad	64K	Resource Manager
$A99C	CountResources	64K	Resource Manager
$A99D	GetIndResource	64K	Resource Manager
$A99E	CountTypes	64K	Resource Manager
$A99F	GetIndType	64K	Resource Manager
$A9A0	GetResource	64K	Resource Manager
$A9A1	GetNamedResource	64K	Resource Manager
$A9A2	LoadResource	64K	Resource Manager
$A9A3	ReleaseResource	64K	Resource Manager

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A9A4	HomeResFile	64K	Resource Manager
$A9A5	SizeResource	64K	Resource Manager
$A9A6	GetResAttrs	64K	Resource Manager
$A9A7	SetResAttrs	64K	Resource Manager
$A9A8	GetResInfo	64K	Resource Manager
$A9A9	SetResInfo	64K	Resource Manager
$A9AA	ChangedResource	64K	Resource Manager
$A9AB	AddResource	64K	Resource Manager
$A9AC	AddReference	64K	Resource Manager
$A9AD	RmveResource	64K	Resource Manager
$A9AE	RmveReference	64K	Resource Manager
$A9AF	ResError	64K	Resource Manager
$A9B0	WriteResource	64K	Resource Manager
$A9B1	CreateResFile	64K	Resource Manager
$A9B2	SystemEvent	64K	Desk Manager
$A9B3	SystemClick	64K	Desk Manager
$A9B4	SystemTask	64K	Desk Manager
$A9B5	SystemMenu	64K	Desk Manager
$A9B6	OpenDeskAcc	64K	Desk Manager
$A9B7	CloseDeskAcc	64K	Desk Manager
$A9B8	GetPattern	64K	Toolbox Utilities
$A9B9	GetCursor	64K	Toolbox Utilities
$A9BA	GetString	64K	Toolbox Utilities
$A9BB	GetIcon	64K	Toolbox Utilities
$A9BC	GetPicture	64K	Toolbox Utilities
$A9BD	GetNewWindow	64K	Window Manager
$A9BE	GetNewControl	64K	Control Manager
$A9BF	GetMenu	64K	Menu Manager
$A9C0	GetNewMBar	64K	Menu Manager
$A9C1	UniqueID	64K	Resource Manager
$A9C2	SystemEdit	64K	Desk Manager
$A9C3	KeyTrans	256K	Toolbox Event Manager
$A9C4	OpenRFPerm	128K	Resource Manager
$A9C5	RsrcMapEntry	128K	Resource Manager
$A9C6	Secs2Date	64K	OS Utilities
$A9C7	Date2Secs	64K	OS Utilities
$A9C8	SysBeep	64K	OS Utilities
$A9C9	SysError	64K	System Error Handler
$A9CA	PutIcon	64K	Toolbox Utilities
$A9CB	TEGetText	64K	TextEdit
$A9CC	TEInit	64K	TextEdit
$A9CD	TEDispose	64K	TextEdit
$A9CE	TextBox	64K	TextEdit
$A9CF	TESetText	64K	TextEdit
$A9D0	TECalText	64K	TextEdit
$A9D1	TESetSelect	64K	TextEdit
$A9D2	TENew	64K	TextEdit
$A9D3	TEUpdate	64K	TextEdit
$A9D4	TEClick	64K	TextEdit
$A9D5	TECopy	64K	TextEdit
$A9D6	TECut	64K	TextEdit

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A9D7	TEDelete	64K	TextEdit
$A9D8	TEActivate	64K	TextEdit
$A9D9	TEDeactivate	64K	TextEdit
$A9DA	TEIdle	64K	TextEdit
$A9DB	TEPaste	64K	TextEdit
$A9DC	TEKey	64K	TextEdit
$A9DD	TEScroll	64K	TextEdit
$A9DE	TEInsert	64K	TextEdit
$A9DF	TESetJust	64K	TextEdit
$A9E0	Munger	64K	Toolbox Utilities
$A9E1	HandToHand	64K	OS Utilities
$A9E2	PtrToXHand	64K	OS Utilities
$A9E3	PtrToHand	64K	OS Utilities
$A9E4	HandAndHand	64K	OS Utilities
$A9E5	InitPack	64K	Package Manager
$A9E6	InitAllPacks	64K	Package Manager
$A9E7	Pack0	64K	Package Manager
$A9E8	Pack1	64K	Package Manager
$A9E9	Pack2 (DiskInit)	64K	Package Manager
$A9EA	Pack3 (StdFile)	64K	Package Manager
$A9EB	Pack4 (Floating Point)	64K	Package Manager
$A9EC	Pack5 (Transcendental Functions)	64K	Package Manager
$A9ED	Pack6 (IntUtils)	64K	Package Manager
$A9EE	Pack7 (Binary-Decimal Conversion)	64K	Package Manager
$A9EF	PtrAndHand	64K	OS Utilities
$A9F0	LoadSeg	64K	Segment Loader
$A9F1	UnloadSeg	64K	Segment Loader
$A9F2	Launch	64K	Segment Loader
$A9F3	Chain	64K	Segment Loader
$A9F4	ExitToShell	64K	Segment Loader
$A9F5	GetAppParms	64K	Segment Loader
$A9F6	GetResFileAttrs	64K	Resource Manager
$A9F7	SetResFileAttrs	64K	Resource Manager
$A9F8	RESERVED	-	-
$A9F9	InfoScrap	64K	Scrap Manager
$A9FA	UnloadScrap	64K	Scrap Manager
$A9FB	LoadScrap	64K	Scrap Manager
$A9FC	ZeroScrap	64K	Scrap Manager
$A9FD	GetScrap	64K	Scrap Manager
$A9FE	PutScrap	64K	Scrap Manager
$A9FF	Debugger	64K	OS Utilities
$AA00	OpenCPort	256K	Color QuickDraw
$AA01	InitCPort	256K	Color QuickDraw
$AA02	RESERVED	-	-
$AA03	NewPixMap	256K	Color QuickDraw
$AA04	DisposPixMap	256K	Color QuickDraw
$AA05	CopyPixMap	256K	Color QuickDraw
$AA06	SetPortPix	256K	Color QuickDraw
$AA07	NewPixPat	256K	Color QuickDraw
$AA08	DisposPixPat	256K	Color QuickDraw
$AA09	CopyPixPat	256K	Color QuickDraw

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$AA0A	PenPixPat	256K	Color QuickDraw
$AA0B	BackPixPat	256K	Color QuickDraw
$AA0C	GetPixPat	256K	Color QuickDraw
$AA0D	MakeRGBPat	256K	Color QuickDraw
$AA0E	FillCRect	256K	Color QuickDraw
$AA0F	FillCOval	256K	Color QuickDraw
$AA10	FillCRoundRect	256K	Color QuickDraw
$AA11	FillCArc	256K	Color QuickDraw
$AA12	FillCRgn	256K	Color QuickDraw
$AA13	FillCPoly	256K	Color QuickDraw
$AA14	RGBForeColor	256K	Color QuickDraw
$AA15	RGBBackColor	256K	Color QuickDraw
$AA16	SetCPixel	256K	Color QuickDraw
$AA17	GetCPixel	256K	Color QuickDraw
$AA18	GetCTable	256K	Color QuickDraw
$AA19	GetForeColor	256K	Color QuickDraw
$AA1A	GetBackColor	256K	Color QuickDraw
$AA1B	GetCCursor	256K	Color QuickDraw
$AA1C	SetCCursor	256K	Color QuickDraw
$AA1D	AllocCursor	256K	Color QuickDraw
$AA1E	GetCIcon	256K	Color QuickDraw
$AA1F	PlotCIcon	256K	Color QuickDraw
$AA20	RESERVED	-	-
$AA21	OpColor	256K	Color QuickDraw
$AA22	HiliteColor	256K	Color QuickDraw
$AA23	CharExtra	256K	Color QuickDraw
$AA24	DisposCTable	256K	Color QuickDraw
$AA25	DisposCIcon	256K	Color QuickDraw
$AA26	DisposCCursor	256K	Color QuickDraw
$AA27	GetMaxDevice	256K	Graphics Devices
$AA28	GetCTSeed	256K	Color Manager
$AA29	GetDeviceList	256K	Graphics Devices
$AA2A	GetMainDevice	256K	Graphics Devices
$AA2B	GetNextDevice	256K	Graphics Devices
$AA2C	TestDeviceAttribute	256K	Graphics Devices
$AA2D	SetDeviceAttribute	256K	Graphics Devices
$AA2E	InitGDevice	256K	Graphics Devices
$AA2F	NewGDevice	256K	Graphics Devices
$AA30	DisposGDevice	256K	Graphics Devices
$AA31	SetGDevice	256K	Graphics Devices
$AA32	GetGDevice	256K	Graphics Devices
$AA33	Color2Index	256K	Color Manager
$AA34	Index2Color	256K	Color Manager
$AA35	InvertColor	256K	Color Manager
$AA36	RealColor	256K	Color Manager
$AA37	GetSubTable	256K	Color Manager
$AA38	RESERVED	-	-
$AA39	MakeITable	256K	Color Manager
$AA3A	AddSearch	256K	Color Manager
$AA3B	AddComp	256K	Color Manager
$AA3C	SetClientID	256K	Color Manager

(continued)

Table 3.004. Continued

Trap Number	*Name*	*ROM **	*Manager*
$AA3D	ProtectEntry	256K	Color Manager
$AA3E	ReserveEntry	256K	Color Manager
$AA3F	SetEntries	256K	Color Manager
$AA40	QDError	256K	Color Manager
$AA41	SetWinColor	256K	Window Manager (color)
$AA42	GetAuxWin	256K	Window Manager (color)
$AA43	SetCtlColor	256K	Control Manager (color)
$AA44	GetAuxCtl	256K	Control Manager (color)
$AA45	NewCWindow	256K	Window Manager (color)
$AA46	GetNewCWindow	256K	Window Manager (color)
$AA47	SetDeskCPat	256K	Window Manager (color)
$AA48	GetCWMgrPort	256K	Window Manager (color)
$AA49	SaveEntries	256K	Color Manager
$AA4A	RestoreEntries	256K	Color Manager
$AA4B	NewCDialog	256K	Dialog Manager (color)
$AA4C	DelSearch	256K	Color Manager
$AA4D	DelComp	256K	Color Manager
$AA4E	SetStdCProcs	256K	Color QuickDraw
$AA4F	CalcCMask	256K	Color QuickDraw
$AA50	SeedCFill	256K	Color QuickDraw
$AA51 to $AA5F	RESERVED	-	-
$AA60	DelMCEntries	256K	Menu Manager (color)
$AA61	GetMCInfo	256K	Menu Manager (color)
$AA62	SetMCInfo	256K	Menu Manager (color)
$AA63	DispMCInfo	256K	Menu Manager (color)
$AA64	GetMCEntry	256K	Menu Manager (color)
$AA65	SetMCEntries	256K	Menu Manager (color)
$AA66	MenuChoice	256K	Menu Manager
$AA67 to $AA8F	RESERVED	-	-
$AA90	InitPalettes	256K	Palette Manager
$AA91	NewPalette	256K	Palette Manager
$AA92	GetNewPalette	256K	Palette Manager
$AA93	DisposPalette	256K	Palette Manager
$AA94	ActivatePalette	256K	Palette Manager
$AA95	SetPalette	256K	Palette Manager
$AA96	GetPalette	256K	Palette Manager
$AA97	PmForeColor	256K	Palette Manager
$AA98	PmBackColor	256K	Palette Manager
$AA99	AnimateEntry	256K	Palette Manager
$AA9A	AnimatePalette	256K	Palette Manager
$AA9B	GetEntryColor	256K	Palette Manager
$AA9C	SetEntryColor	256K	Palette Manager
$AA9D	GetEntryUsage	256K	Palette Manager
$AA9E	SetEntryUsage	256K	Palette Manager
$AA9F	CTab2Palette	256K	Palette Manager
$AAA0	Palette2CTab	256K	Palette Manager
$AAA1 to $ABF7	RESERVED	-	-
$ABF8	StdOpcodeProc	256K	Color Manager
$ABF9 to $ABFE	RESERVED	-	-
$ABFF	DebugStr	64K	OS Utilities
$AC00 to $AFFF	RESERVED	-	-

(continued)

Table 3.004. Continued

Notes:

•* First ROM that includes trap.
•** For internal use only; not used by applications.
•*** Dispatched based upon additional information:

0	FSControl
1	PBOpenWD
2	PBCloseWD
5	PBCatMove
6	PBDirCreate
7	PBGetWDInfo
8	PBGetFCBInfo
9	PBGetCatInfo
10	PBSetCatInfo
11	PBSetVolInfo
16	PBLockRng
17	PBUnlockRng
$30	GetVolParms
$31	GetLogInInfo
$32	GetDirAccess
$33	SetDirAccess
$34	MapID
$35	MapName
$36	CopyFile
$37	MoveRename
$38	OpenDeny
$39	OpenRFDeny

•**** Dispatched based upon additional information:

0	TEStylPaste
1	TESetStyle
2	TEReplaceStyle
3	TEGetStyle
4	GetStylHandle
5	SetStylHandle
6	GetStylScrap
7	TEStylInsert
8	TEGetPoint
9	TEGetHeight

•† Dispatched based upon additional information:

1	ShutDwnPower
2	ShutDwnStart
3	ShutDwnInstall
4	ShutDwnRemove

Source: Inside Macintosh, pages III-215 to III-226, V-603 to V-605, X-Ref-63 to X-Ref-72

See Also: 3.008. Interface Toolbox Managers
3.156. Operating System Calls
And other tables for various Toolbox and OS routines

3.005 System Traps by Manager

Trap Number	*Name*	*ROM **	*Manager*
$A07C	ADBOp	256K	Apple DeskTop Bus
$A07B	ADBReInit	256K	Apple DeskTop Bus
$A077	CountADBs	256K	Apple DeskTop Bus
$A079	GetADBInfo	256K	Apple DeskTop Bus
$A078	GetIndADB	256K	Apple DeskTop Bus
$A07A	SetADBInfo	256K	Apple DeskTop Bus
$A809	GetCVariant	256K	Color Controls
$AA3B	AddComp	256K	Color Manager
$AA3A	AddSearch	256K	Color Manager
$AA33	Color2Index	256K	Color Manager
$AA4D	DelComp	256K	Color Manager
$AA4C	DelSearch	256K	Color Manager
$AA28	GetCTSeed	256K	Color Manager
$AA37	GetSubTable	256K	Color Manager
$AA34	Index2Color	256K	Color Manager
$AA35	InvertColor	256K	Color Manager
$AA39	MakeITable	256K	Color Manager
$AA3D	ProtectEntry	256K	Color Manager
$AA40	QDError	256K	Color Manager
$AA36	RealColor	256K	Color Manager
$AA3E	ReserveEntry	256K	Color Manager
$AA4A	RestoreEntries	256K	Color Manager
$AA49	SaveEntries	256K	Color Manager
$AA3C	SetClientID	256K	Color Manager
$AA3F	SetEntries	256K	Color Manager
$ABF8	StdOpcodeProc	256K	Color Manager
$AA1D	AllocCursor	256K	Color QuickDraw
$AA0B	BackPixPat	256K	Color QuickDraw
$AA4F	CalcCMask	256K	Color QuickDraw
$AA23	CharExtra	256K	Color QuickDraw
$A87D	CloseCPort	256K	Color QuickDraw
$AA05	CopyPixMap	256K	Color QuickDraw
$AA09	CopyPixPat	256K	Color QuickDraw
$AA26	DisposCCursor	256K	Color QuickDraw
$AA25	DisposCIcon	256K	Color QuickDraw
$AA24	DisposCTable	256K	Color QuickDraw
$AA04	DisposPixMap	256K	Color QuickDraw
$AA08	DisposPixPat	256K	Color QuickDraw
$AA11	FillCArc	256K	Color QuickDraw
$AA0F	FillCOval	256K	Color QuickDraw
$AA13	FillCPoly	256K	Color QuickDraw
$AA0E	FillCRect	256K	Color QuickDraw
$AA12	FillCRgn	256K	Color QuickDraw
$AA10	FillCRoundRect	256K	Color QuickDraw
$AA1A	GetBackColor	256K	Color QuickDraw
$AA1B	GetCCursor	256K	Color QuickDraw
$AA1E	GetCIcon	256K	Color QuickDraw
$AA17	GetCPixel	256K	Color QuickDraw
$AA18	GetCTable	256K	Color QuickDraw
$AA19	GetForeColor	256K	Color QuickDraw
$AA0C	GetPixPat	256K	Color QuickDraw

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$AA22	HiliteColor	256K	Color QuickDraw
$AA01	InitCPort	256K	Color QuickDraw
$AA0D	MakeRGBPat	256K	Color QuickDraw
$AA03	NewPixMap	256K	Color QuickDraw
$AA07	NewPixPat	256K	Color QuickDraw
$AA21	OpColor	256K	Color QuickDraw
$AA00	OpenCPort	256K	Color QuickDraw
$AA0A	PenPixPat	256K	Color QuickDraw
$AA1F	PlotCIcon	256K	Color QuickDraw
$AA15	RGBBackColor	256K	Color QuickDraw
$AA14	RGBForeColor	256K	Color QuickDraw
$AA50	SeedCFill	256K	Color QuickDraw
$AA1C	SetCCursor	256K	Color QuickDraw
$AA16	SetCPixel	256K	Color QuickDraw
$AA06	SetPortPix	256K	Color QuickDraw
$AA4E	SetStdCProcs	256K	Color QuickDraw
$A955	DisposeControl	64K	Control Manager
$A967	DragControl	64K	Control Manager
$A96D	Draw1Control	128K	Control Manager
$A969	DrawControls	64K	Control Manager
$A96C	FindControl	64K	Control Manager
$AA44	GetAuxCtl	256K	Control Manager
$A95A	GetCRefCon	64K	Control Manager
$A95E	GetCTitle	64K	Control Manager
$A96A	GetCtlAction	64K	Control Manager
$A962	GetCtlMax	64K	Control Manager
$A961	GetCtlMin	64K	Control Manager
$A960	GetCtlValue	64K	Control Manager
$A9BE	GetNewControl	64K	Control Manager
$A958	HideControl	64K	Control Manager
$A95D	HiliteControl	64K	Control Manager
$A956	KillControls	64K	Control Manager
$A959	MoveControl	64K	Control Manager
$A954	NewControl	64K	Control Manager
$A95B	SetCRefCon	64K	Control Manager
$A95F	SetCTitle	64K	Control Manager
$A96B	SetCtlAction	64K	Control Manager
$AA43	SetCtlColor	256K	Control Manager
$A965	SetCtlMax	64K	Control Manager
$A964	SetCtlMin	64K	Control Manager
$A963	SetCtlValue	64K	Control Manager
$A957	ShowControl	64K	Control Manager
$A95C	SizeControl	64K	Control Manager
$A966	TestControl	64K	Control Manager
$A968	TrackControl	64K	Control Manager
$A953	UpdtControl	128K	Control Manager
$A9B7	CloseDeskAcc	64K	Desk Manager
$A9B6	OpenDeskAcc	64K	Desk Manager
$A9B3	SystemClick	64K	Desk Manager
$A9C2	SystemEdit	64K	Desk Manager
$A9B2	SystemEvent	64K	Desk Manager

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A9B5	SystemMenu	64K	Desk Manager
$A9B4	SystemTask	64K	Desk Manager
$A04E	AddDrive **	64K	Device Manager
$A004	Control	64K	Device Manager
$A006	KillIO	64K	Device Manager
$A04F	RDrvrInstall **	64K	Device Manager
$A075	SIntInstall	256K	Device Manager
$A076	SIntRemove	256K	Device Manager
$A005	Status	64K	Device Manager
$A985	Alert	64K	Dialog Manager
$A988	CautionAlert	64K	Dialog Manager
$A982	CloseDialog	64K	Dialog Manager
$A989	CouldAlert	64K	Dialog Manager
$A979	CouldDialog	64K	Dialog Manager
$A980	DialogSelect	64K	Dialog Manager
$A983	DisposDialog	64K	Dialog Manager
$A981	DrawDialog	64K	Dialog Manager
$A98C	ErrorSound	64K	Dialog Manager
$A984	FindDItem	128K	Dialog Manager
$A98A	FreeAlert	64K	Dialog Manager
$A97A	FreeDialog	64K	Dialog Manager
$A98D	GetDItem	64K	Dialog Manager
$A990	GetIText	64K	Dialog Manager
$A97C	GetNewDialog	64K	Dialog Manager
$A827	HideDItem	128K	Dialog Manager
$A97B	InitDialogs	64K	Dialog Manager
$A97F	IsDialogEvent	64K	Dialog Manager
$A991	ModalDialog	64K	Dialog Manager
$AA4B	NewCDialog	256K	Dialog Manager
$A97D	NewDialog	64K	Dialog Manager
$A987	NoteAlert	64K	Dialog Manager
$A98B	ParamText	64K	Dialog Manager
$A97E	SelIText	64K	Dialog Manager
$A98E	SetDItem	64K	Dialog Manager
$A98F	SetIText	64K	Dialog Manager
$A828	ShowDItem	128K	Dialog Manager
$A986	StopAlert	64K	Dialog Manager
$A978	UpdtDialog	128K	Dialog Manager
$A010	Allocate	64K	File Manager
$A20F	AllocContig	128K	File Manager
$A008	Create	64K	File Manager
$A009	Delete	64K	File Manager
$A017	Eject	64K	File Manager
$A045	FlushFile	64K	File Manager
$A013	FlushVol	64K	File Manager
$A011	GetEOF	64K	File Manager
$A00C	GetFileInfo	64K	File Manager
$A018	GetFPos	64K	File Manager
$A014	GetVol	64K	File Manager
$A007	GetVolInfo	64K	File Manager
$A208	HCreate	128K	File Manager

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A209	HDelete	128K	File Manager
$A260	HFSDispatch ***	128K	File Manager
$A20B	HGetFileInfo	128K	File Manager
$A214	HGetVol	128K	File Manager
$A207	HGetWDInfo	128K	File Manager
$A200	HOpen	128K	File Manager
$A20A	HOpenRF	128K	File Manager
$A20C	HRename	128K	File Manager
$A242	HRstFLock	128K	File Manager
$A20E	HSetFileInfo	128K	File Manager
$A241	HSetFLock	128K	File Manager
$A215	HSetVol	128K	File Manager
$A06C	InitFS	128K	File Manager
$A016	InitQueue	64K	File Manager
$A00F	MountVol	64K	File Manager
$A035	Offline	64K	File Manager
$A000	Open	64K	File Manager
$A00A	OpenRF	64K	File Manager
$A001	PBClose	64K	File Manager
$A002	Read	64K	File Manager
$A00B	Rename	64K	File Manager
$A042	RstFilLock	64K	File Manager
$A012	SetEOF	64K	File Manager
$A00D	SetFileInfo	64K	File Manager
$A041	SetFilLock	64K	File Manager
$A043	SetFilType	64K	File Manager
$A044	SetFPos	64K	File Manager
$A015	SetVol	64K	File Manager
$A00E	UnmountVol	64K	File Manager
$A003	Write	64K	File Manager
$A901	FMSwapFont	64K	Font Manager
$A835	FontMetrics	128K	Font Manager
$A900	GetFNum	64K	Font Manager
$A8FF	GetFontName	64K	Font Manager
$A8FE	InitFonts	64K	Font Manager
$A902	RealFont	64K	Font Manager
$A903	SetFontLock	64K	Font Manager
$A814	SetFractEnable	256K	Font Manager
$A834	SetFScaleDisable	128K	Font Manager
$AA30	DisposGDevice	256K	Graphics Devices
$AA29	GetDeviceList	256K	Graphics Devices
$AA32	GetGDevice	256K	Graphics Devices
$AA2A	GetMainDevice	256K	Graphics Devices
$AA27	GetMaxDevice	256K	Graphics Devices
$AA2B	GetNextDevice	256K	Graphics Devices
$AA2E	InitGDevice	256K	Graphics Devices
$AA2F	NewGDevice	256K	Graphics Devices
$AA2D	SetDeviceAttribute	256K	Graphics Devices
$AA31	SetGDevice	256K	Graphics Devices
$AA2C	TestDeviceAttribute	256K	Graphics Devices
$A02E	BlockMove	64K	Memory Manager

(continued)

Table 3.005. Continued

Trap Number	Name	ROM *	Manager
$A04C	CompactMem	64K	Memory Manager
$A023	DisposHandle	64K	Memory Manager
$A01F	DisposPtr	64K	Memory Manager
$A02B	EmptyHandle	64K	Memory Manager
$A01C	FreeMem	64K	Memory Manager
$A025	GetHandleSize	64K	Memory Manager
$A021	GetPtrSize	64K	Memory Manager
$A11A	GetZone	64K	Memory Manager
$A126	HandleZone	64K	Memory Manager
$A068	HClrRBit	128K	Memory Manager
$A069	HGetState	128K	Memory Manager
$A029	HLock	64K	Memory Manager
$A04A	HNoPurge	64K	Memory Manager
$A049	HPurge	64K	Memory Manager
$A067	HSetRBit	128K	Memory Manager
$A06A	HSetState	128K	Memory Manager
$A02A	HUnlock	64K	Memory Manager
$A02C	InitApplZone	64K	Memory Manager
$A019	InitZone	64K	Memory Manager
$A063	MaxApplZone	128K	Memory Manager
$A061	MaxBlock	128K	Memory Manager
$A11D	MaxMem	64K	Memory Manager
$A036	MoreMasters	64K	Memory Manager
$A064	MoveHHi	128K	Memory Manager
$A166	NewEmptyHandle	128K	Memory Manager
$A122	NewHandle	64K	Memory Manager
$A11E	NewPtr	64K	Memory Manager
$A048	PtrZone	64K	Memory Manager
$A148	PtrZone	64K	Memory Manager
$A04D	PurgeMem	64K	Memory Manager
$A062	PurgeSpace	128K	Memory Manager
$A027	ReallocHandle	64K	Memory Manager
$A128	RecoverHandle	64K	Memory Manager
$A040	ResrvMem	64K	Memory Manager
$A057	SetAppBase	64K	Memory Manager
$A02D	SetApplLimit	64K	Memory Manager
$A04B	SetGrowZone	64K	Memory Manager
$A024	SetHandleSize	64K	Memory Manager
$A020	SetPtrSize	64K	Memory Manager
$A01B	SetZone	64K	Memory Manager
$A065	StackSpace	128K	Memory Manager
$A94D	AddResMenu	64K	Menu Manager
$A933	AppendMenu	64K	Menu Manager
$A948	CalcMenuSize	64K	Menu Manager
$A945	CheckItem	64K	Menu Manager
$A934	ClearMenuBar	64K	Menu Manager
$A950	CountMItems	64K	Menu Manager
$A936	DeleteMenu	64K	Menu Manager
$AA60	DelMCEntries	256K	Menu Manager
$A952	DelMenuItem	128K	Menu Manager
$A93A	DisableItem	64K	Menu Manager

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$AA63	DispMCInfo	256K	Menu Manager
$A932	DisposeMenu	64K	Menu Manager
$A937	DrawMenuBar	64K	Menu Manager
$A939	EnableItem	64K	Menu Manager
$A94C	FlashMenuBar	64K	Menu Manager
$A946	GetItem	64K	Menu Manager
$A84E	GetItemCmd	256K	Menu Manager
$A93F	GetItemIcon	64K	Menu Manager
$A943	GetItemMark	64K	Menu Manager
$A941	GetItemStyle	64K	Menu Manager
$AA64	GetMCEntry	256K	Menu Manager
$AA61	GetMCInfo	256K	Menu Manager
$A9BF	GetMenu	64K	Menu Manager
$A93B	GetMenuBar	64K	Menu Manager
$A949	GetMHandle	64K	Menu Manager
$A9C0	GetNewMBar	64K	Menu Manager
$A938	HiliteMenu	64K	Menu Manager
$A930	InitMenus	64K	Menu Manager
$A808	InitProcMenu	256K	Menu Manager
$A935	InsertMenu	64K	Menu Manager
$A951	InsertResMenu	64K	Menu Manager
$A826	InsMenuItem	128K	Menu Manager
$AA66	MenuChoice	256K	Menu Manager
$A93E	MenuKey	64K	Menu Manager
$A93D	MenuSelect	64K	Menu Manager
$A931	NewMenu	64K	Menu Manager
$A80B	PopUpMenuSelect	256K	Menu Manager
$A947	SetItem	64K	Menu Manager
$A84F	SetItemCmd	256K	Menu Manager
$A940	SetItemIcon	64K	Menu Manager
$A944	SetItemMark	64K	Menu Manager
$A942	SetItemStyle	64K	Menu Manager
$AA65	SetMCEntries	256K	Menu Manager
$AA62	SetMCInfo	256K	Menu Manager
$A93C	SetMenuBar	64K	Menu Manager
$A94A	SetMenuFlash	64K	Menu Manager
$A032	FlushEvents	64K	OS Event Manager
$A030	OSEventAvail	64K	OS Event Manager
$A02F	PostEvent	64K	OS Event Manager
$A12F	PPostEvent	128K	OS Event Manager
$A973	StillDown	64K	OS Event Manager
$A975	TickCount	64K	OS Event Manager
$A977	WaitMouseUp	64K	OS Event Manager
$A9C7	Date2Secs	64K	OS Utilities
$A9FF	Debugger	64K	OS Utilities
$ABFF	DebugStr	64K	OS Utilities
$A03B	Delay	64K	OS Utilities
$A96E	Dequeue	64K	OS Utilities
$A03D	DrvrInstall **	64K	OS Utilities
$A03E	DrvrRemove **	64K	OS Utilities
$A96F	Enqueue	64K	OS Utilities

(continued)

Table 3.005. Continued

Trap Number	Name	ROM *	Manager
$A03C	EqualString	64K	OS Utilities
$A146	GetTrapAddress	64K	OS Utilities
$A9E4	HandAndHand	64K	OS Utilities
$A9E1	HandToHand	64K	OS Utilities
$A03F	InitUtil	64K	OS Utilities
$A9EF	PtrAndHand	64K	OS Utilities
$A9E3	PtrToHand	64K	OS Utilities
$A9E2	PtrToXHand	64K	OS Utilities
$A039	ReadDateTime	64K	OS Utilities
$A037	ReadParam	64K	OS Utilities
$A051	ReadXPRam	128K	OS Utilities
$A050	RelString	128K	OS Utilities
$A833	ScrnBitMap	128K	OS Utilities
$A9C6	Secs2Date	64K	OS Utilities
$A03A	SetDateTime	64K	OS Utilities
$A047	SetTrapAddress	64K	OS Utilities
$A055	StripAddress	256K	OS Utilities
$A05D	SwapMMUMode	256K	OS Utilities
$A9C8	SysBeep	64K	OS Utilities
$A090	SysEnvirons	256K	OS Utilities
$A054	UprString	64K	OS Utilities
$A038	WriteParam	64K	OS Utilities
$A052	WriteXPRam	128K	OS Utilities
$A9E6	InitAllPacks	64K	Package Manager
$A9E5	InitPack	64K	Package Manager
$A9E7	Pack0	128K	Package Manager
$A9E8	Pack1	64K	Package Manager
$A82C	Pack10	128K	Package Manager
$A82D	Pack11	128K	Package Manager
$A82E	Pack12 (ColorPicker)	128K	Package Manager
$A82F	Pack13	128K	Package Manager
$A830	Pack14	128K	Package Manager
$A831	Pack15	128K	Package Manager
$A9E9	Pack2 (DiskInit)	64K	Package Manager
$A9EA	Pack3 (StdFile)	64K	Package Manager
$A9EB	Pack4 (Floating Point)	64K	Package Manager
$A9EC	Pack5 (Transcendental Functions)	64K	Package Manager
$A9ED	Pack6 (IntUtils)	64K	Package Manager
$A9EE	Pack7 (Binary-Decimal Conversion)	64K	Package Manager
$A816	Pack8	128K	Package Manager
$A82B	Pack9	128K	Package Manager
$AA94	ActivatePalette	256K	Palette Manager
$AA99	AnimateEntry	256K	Palette Manager
$AA9A	AnimatePalette	256K	Palette Manager
$AA9F	CTab2Palette	256K	Palette Manager
$AA93	DisposPalette	256K	Palette Manager
$AA9B	GetEntryColor	256K	Palette Manager
$AA9D	GetEntryUsage	256K	Palette Manager
$AA92	GetNewPalette	256K	Palette Manager
$AA96	GetPalette	256K	Palette Manager
$AA90	InitPalettes	256K	Palette Manager

(continued)

Table 3.005. Continued

Trap Number	Name	ROM *	Manager
$AA91	NewPalette	256K	Palette Manager
$AAA0	Palette2CTab	256K	Palette Manager
$AA98	PmBackColor	256K	Palette Manager
$AA97	PmForeColor	256K	Palette Manager
$AA9C	SetEntryColor	256K	Palette Manager
$AA9E	SetEntryUsage	256K	Palette Manager
$AA95	SetPalette	256K	Palette Manager
$A8FD	PrGlue	256K	Printing Manager
$A87E	AddPt	64K	QuickDraw
$A863	BackColor	64K	QuickDraw
$A87C	BackPat	64K	QuickDraw
$A838	CalcMask	128K	QuickDraw
$A88D	CharWidth	64K	QuickDraw
$A87B	ClipRect	64K	QuickDraw
$A8F4	ClosePicture	64K	QuickDraw
$A8CC	ClosePoly	64K	QuickDraw
$A87D	ClosePort	64K	QuickDraw
$A8DB	CloseRgn	64K	QuickDraw
$A864	ColorBit	64K	QuickDraw
$A8EC	CopyBits	64K	QuickDraw
$A817	CopyMask	128K	QuickDraw
$A8DC	CopyRgn	64K	QuickDraw
$A8E6	DiffRgn	64K	QuickDraw
$A8D9	DisposeRgn	64K	QuickDraw
$A883	DrawChar	64K	QuickDraw
$A8F6	DrawPicture	64K	QuickDraw
$A884	DrawString	64K	QuickDraw
$A885	DrawText	64K	QuickDraw
$A8AE	EmptyRect	64K	QuickDraw
$A8E2	EmptyRgn	64K	QuickDraw
$A881	EqualPt	64K	QuickDraw
$A8A6	EqualRect	64K	QuickDraw
$A8E3	EqualRgn	64K	QuickDraw
$A8C0	EraseArc	64K	QuickDraw
$A8B9	EraseOval	64K	QuickDraw
$A8C8	ErasePoly	64K	QuickDraw
$A8A3	EraseRect	64K	QuickDraw
$A8D4	EraseRgn	64K	QuickDraw
$A8B2	EraseRoundRect	64K	QuickDraw
$A8C2	FillArc	64K	QuickDraw
$A8BB	FillOval	64K	QuickDraw
$A8CA	FillPoly	64K	QuickDraw
$A8A5	FillRect	64K	QuickDraw
$A8D6	FillRgn	64K	QuickDraw
$A8B4	FillRoundRect	64K	QuickDraw
$A862	ForeColor	64K	QuickDraw
$A8BE	FrameArc	64K	QuickDraw
$A8B7	FrameOval	64K	QuickDraw
$A8C6	FramePoly	64K	QuickDraw
$A8A1	FrameRect	64K	QuickDraw
$A8D2	FrameRgn	64K	QuickDraw

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A8B0	FrameRoundRect	64K	QuickDraw
$A87A	GetClip	64K	QuickDraw
$A88B	GetFontInfo	64K	QuickDraw
$A836	GetMaskTable	128K	QuickDraw
$A89A	GetPen	64K	QuickDraw
$A898	GetPenState	64K	QuickDraw
$A865	GetPixel	64K	QuickDraw
$A874	GetPort	64K	QuickDraw
$A871	GlobalToLocal	64K	QuickDraw
$A872	GrafDevice	64K	QuickDraw
$A852	HideCursor	64K	QuickDraw
$A896	HidePen	64K	QuickDraw
$A850	InitCursor	64K	QuickDraw
$A86E	InitGraf	64K	QuickDraw
$A86D	InitPort	64K	QuickDraw
$A8A9	InsetRect	64K	QuickDraw
$A8E1	InsetRgn	64K	QuickDraw
$A8C1	InvertArc	64K	QuickDraw
$A8BA	InvertOval	64K	QuickDraw
$A8C9	InvertPoly	64K	QuickDraw
$A8A4	InvertRect	64K	QuickDraw
$A8D5	InvertRgn	64K	QuickDraw
$A8B3	InvertRoundRect	64K	QuickDraw
$A8F5	KillPicture	64K	QuickDraw
$A8CD	KillPoly	64K	QuickDraw
$A892	Line	64K	QuickDraw
$A891	LineTo	64K	QuickDraw
$A870	LocalToGlobal	64K	QuickDraw
$A8FC	MapPoly	64K	QuickDraw
$A8F9	MapPt	64K	QuickDraw
$A8FA	MapRect	64K	QuickDraw
$A8FB	MapRgn	64K	QuickDraw
$A837	MeasureText	128K	QuickDraw
$A894	Move	64K	QuickDraw
$A877	MovePortTo	64K	QuickDraw
$A893	MoveTo	64K	QuickDraw
$A8D8	NewRgn	64K	QuickDraw
$A856	ObscureCursor	64K	QuickDraw
$A8CE	OffsetPoly	64K	QuickDraw
$A8A8	OffsetRect	64K	QuickDraw
$A8E0	OffsetRgn	64K	QuickDraw
$A8F3	OpenPicture	64K	QuickDraw
$A8CB	OpenPoly	64K	QuickDraw
$A86F	OpenPort	64K	QuickDraw
$A8DA	OpenRgn	64K	QuickDraw
$A8BF	PaintArc	64K	QuickDraw
$A8B8	PaintOval	64K	QuickDraw
$A8C7	PaintPoly	64K	QuickDraw
$A8A2	PaintRect	64K	QuickDraw
$A8D3	PaintRgn	64K	QuickDraw
$A8B1	PaintRoundRect	64K	QuickDraw

(continued)

Table 3.005. Continued

Trap Number	Name	ROM *	Manager
$A89C	PenMode	64K	QuickDraw
$A89E	PenNormal	64K	QuickDraw
$A89D	PenPat	64K	QuickDraw
$A89B	PenSize	64K	QuickDraw
$A8F2	PicComment	64K	QuickDraw
$A876	PortSize	64K	QuickDraw
$A8AC	Pt2Rect	64K	QuickDraw
$A8AD	PtInRect	64K	QuickDraw
$A8E8	PtInRgn	64K	QuickDraw
$A8C3	PtToAngle	64K	QuickDraw
$A861	Random	64K	QuickDraw
$A8E9	RectInRgn	64K	QuickDraw
$A8DF	RectRgn	64K	QuickDraw
$A8F8	ScalePt	64K	QuickDraw
$A8EF	ScrollRect	64K	QuickDraw
$A8AA	SectRect	64K	QuickDraw
$A8E4	SectRgn	64K	QuickDraw
$A839	SeedFill	128K	QuickDraw
$A879	SetClip	64K	QuickDraw
$A851	SetCursor	64K	QuickDraw
$A8DD	SetEmptyRgn	64K	QuickDraw
$A878	SetOrigin	64K	QuickDraw
$A899	SetPenState	64K	QuickDraw
$A873	SetPort	64K	QuickDraw
$A875	SetPortBits	64K	QuickDraw
$A880	SetPt	64K	QuickDraw
$A8A7	SetRect	64K	QuickDraw
$A8DE	SetRectRgn	64K	QuickDraw
$A8EA	SetStdProcs	64K	QuickDraw
$A853	ShowCursor	64K	QuickDraw
$A897	ShowPen	64K	QuickDraw
$A88E	SpaceExtra	64K	QuickDraw
$A8BD	StdArc	64K	QuickDraw
$A8EB	StdBits	64K	QuickDraw
$A8F1	StdComment	64K	QuickDraw
$A8EE	StdGetPic	64K	QuickDraw
$A890	StdLine	64K	QuickDraw
$A8B6	StdOval	64K	QuickDraw
$A8C5	StdPoly	64K	QuickDraw
$A8F0	StdPutPic	64K	QuickDraw
$A8A0	StdRect	64K	QuickDraw
$A8D1	StdRgn	64K	QuickDraw
$A8AF	StdRRect	64K	QuickDraw
$A882	StdText	64K	QuickDraw
$A8ED	StdTxMeas	64K	QuickDraw
$A88C	StringWidth	64K	QuickDraw
$A866	StuffHex	64K	QuickDraw
$A87F	SubPt	64K	QuickDraw
$A888	TextFace	64K	QuickDraw
$A887	TextFont	64K	QuickDraw
$A889	TextMode	64K	QuickDraw

(continued)

Table 3.005. Continued

Trap Number	Name	ROM *	Manager
$A88A	TextSize	64K	QuickDraw
$A886	TextWidth	64K	QuickDraw
$A8AB	UnionRect	64K	QuickDraw
$A8E5	UnionRgn	64K	QuickDraw
$A8E7	XorRgn	64K	QuickDraw
$A9AC	AddReference	64K	Resource Manager
$A9AB	AddResource	64K	Resource Manager
$A9AA	ChangedResource	64K	Resource Manager
$A99A	CloseResFile	64K	Resource Manager
$A80D	Count1Resources	128K	Resource Manager
$A81C	Count1Types	128K	Resource Manager
$A99C	CountResources	64K	Resource Manager
$A99E	CountTypes	64K	Resource Manager
$A9B1	CreateResFile	64K	Resource Manager
$A994	CurResFile	64K	Resource Manager
$A992	DetachResource	64K	Resource Manager
$A80E	Get1IndResource	128K	Resource Manager
$A80F	Get1IndType	128K	Resource Manager
$A820	Get1NamedResource	128K	Resource Manager
$A81F	Get1Resource	128K	Resource Manager
$A99D	GetIndResource	64K	Resource Manager
$A99F	GetIndType	64K	Resource Manager
$A9A1	GetNamedResource	64K	Resource Manager
$A9A6	GetResAttrs	64K	Resource Manager
$A9F6	GetResFileAttrs	64K	Resource Manager
$A9A8	GetResInfo	64K	Resource Manager
$A9A0	GetResource	64K	Resource Manager
$A9A4	HomeResFile	64K	Resource Manager
$A995	InitResources	64K	Resource Manager
$A9A2	LoadResource	64K	Resource Manager
$A821	MaxSizeRsrc	128K	Resource Manager
$A997	OpenResFile	64K	Resource Manager
$A9C4	OpenRFPerm	128K	Resource Manager
$A9A3	ReleaseResource	64K	Resource Manager
$A9AF	ResError	64K	Resource Manager
$A80C	RGetResource	256K	Resource Manager
$A9AE	RmveReference	64K	Resource Manager
$A9AD	RmveResource	64K	Resource Manager
$A9C5	RsrcMapEntry	128K	Resource Manager
$A996	RsrcZoneInit	64K	Resource Manager
$A9A7	SetResAttrs	64K	Resource Manager
$A9F7	SetResFileAttrs	64K	Resource Manager
$A9A9	SetResInfo	64K	Resource Manager
$A99B	SetResLoad	64K	Resource Manager
$A993	SetResPurge	64K	Resource Manager
$A9A5	SizeResource	64K	Resource Manager
$A810	Unique1ID	128K	Resource Manager
$A9C1	UniqueID	64K	Resource Manager
$A999	UpdateResFile	64K	Resource Manager
$A998	UseResFile	64K	Resource Manager
$A9B0	WriteResource	64K	Resource Manager

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A9FD	GetScrap	64K	Scrap Manager
$A9F9	InfoScrap	64K	Scrap Manager
$A9FB	LoadScrap	64K	Scrap Manager
$A9FE	PutScrap	64K	Scrap Manager
$A9FA	UnloadScrap	64K	Scrap Manager
$A9FC	ZeroScrap	64K	Scrap Manager
$A8B5	ScriptUtil	256K	Script Manager
$A815	SCSIDispatch	128K	SCSI Manager
$A9F3	Chain	64K	Segment Loader
$A9F4	ExitToShell	64K	Segment Loader
$A9F5	GetAppParms	64K	Segment Loader
$A9F2	Launch	64K	Segment Loader
$A9F0	LoadSeg	64K	Segment Loader
$A9F1	UnloadSeg	64K	Segment Loader
$A895	ShutDown Dispatch ****	256K	Shutdown Manager
$A06E	SlotManager	256K	Slot Manager
$A802	SndAddModifier	256K	Sound Manager
$A806	SndControl	256K	Sound Manager
$A803	SndDoCommand	256K	Sound Manager
$A804	SndDoImmediate	256K	Sound Manager
$A807	SndNewChannel	256K	Sound Manager
$A805	SndPlay	256K	Sound Manager
$A07D	GetDefaultStartup	256K	Start Manager
$A084	GetOSDefault	256K	Start Manager
$A080	GetVideoDefault	256K	Start Manager
$A07F	InternalWait	256K	Start Manager
$A07E	SetDefaultStartup	256K	Start Manager
$A083	SetOSDefault	256K	Start Manager
$A081	SetVideoDefault	256K	Start Manager
$A9C9	SysError	64K	System Error Handler
$A9D8	TEActivate	64K	TextEdit
$A813	TEAutoView	128K	TextEdit
$A9D0	TECalText	64K	TextEdit
$A9D4	TEClick	64K	TextEdit
$A9D5	TECopy	64K	TextEdit
$A9D6	TECut	64K	TextEdit
$A9D9	TEDeactivate	64K	TextEdit
$A9D7	TEDelete	64K	TextEdit
$A83D	TEDispatch †	256K	TextEdit
$A9CD	TEDispose	64K	TextEdit
$A83C	TEGetOffset	256K	TextEdit
$A9CB	TEGetText	64K	TextEdit
$A9DA	TEIdle	64K	TextEdit
$A9CC	TEInit	64K	TextEdit
$A9DE	TEInsert	64K	TextEdit
$A9DC	TEKey	64K	TextEdit
$A9D2	TENew	64K	TextEdit
$A9DB	TEPaste	64K	TextEdit
$A812	TEPinScroll	128K	TextEdit
$A9DD	TEScroll	64K	TextEdit
$A811	TESelView	128K	TextEdit

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A9DF	TESetJust	64K	TextEdit
$A9D1	TESetSelect	64K	TextEdit
$A9CF	TESetText	64K	TextEdit
$A83E	TEStyleNew	256K	TextEdit
$A9D3	TEUpdate	64K	TextEdit
$A9CE	TextBox	64K	TextEdit
$A058	InsTime	128K	Time Manager
$A05A	PrimeTime	128K	Time Manager
$A059	RmvTime	128K	Time Manager
$A974	Button	64K	Toolbox Event Manager
$A971	EventAvail	64K	Toolbox Event Manager
$A976	GetKeys	64K	Toolbox Event Manager
$A972	GetMouse	64K	Toolbox Event Manager
$A970	GetNextEvent	64K	Toolbox Event Manager
$A031	GetOSEvent	64K	Toolbox Event Manager
$A9C3	KeyTrans	256K	Toolbox Event Manager
$A8C4	AngleFromSlope	64K	Toolbox Utilities
$A85F	BitClr	64K	Toolbox Utilities
$A85A	BitNot	64K	Toolbox Utilities
$A85B	BitOr	64K	Toolbox Utilities
$A85E	BitSet	64K	Toolbox Utilities
$A85C	BitShift	64K	Toolbox Utilities
$A85D	BitTst	64K	Toolbox Utilities
$A859	BitXOr	64K	Toolbox Utilities
$A94F	DeltaPoint	64K	Toolbox Utilities
$A818	FixATan2	128K	Toolbox Utilities
$A868	FixMul	64K	Toolbox Utilities
$A869	FixRatio	64K	Toolbox Utilities
$A86D	FixRound	64K	Toolbox Utilities
$A9B9	GetCursor	64K	Toolbox Utilities
$A9BB	GetIcon	64K	Toolbox Utilities
$A9B8	GetPattern	64K	Toolbox Utilities
$A9BC	GetPicture	64K	Toolbox Utilities
$A9BA	GetString	64K	Toolbox Utilities
$A86A	HiWord	64K	Toolbox Utilities
$A867	LongMul	64K	Toolbox Utilities
$A86B	LoWord	64K	Toolbox Utilities
$A9E0	Munger	64K	Toolbox Utilities
$A906	NewString	64K	Toolbox Utilities
$A8CF	PackBits	64K	Toolbox Utilities
$A94B	PlotIcon	64K	Toolbox Utilities
$A9CA	PutIcon	64K	Toolbox Utilities
$A907	SetString	64K	Toolbox Utilities
$A855	ShieldCursor	64K	Toolbox Utilities
$A8BC	SlopeFromAngle	64K	Toolbox Utilities
$A8D0	UnpackBits	64K	Toolbox Utilities
$A819	XMunger	128K	Toolbox Utilities
$A071	AttachVBL	256K	Vertical Retrace Manager
$A072	DoVBLTask	256K	Vertical Retrace Manager
$A06F	SlotVInstall	256K	Vertical Retrace Manager
$A070	SlotVRemove	256K	Vertical Retrace Manager

(continued)

Table 3.005. Continued

Trap Number	*Name*	*ROM **	*Manager*
$A033	VInstall	64K	Vertical Retrace Manager
$A034	VRemove	64K	Vertical Retrace Manager
$A922	BeginUpdate	64K	Window Manager
$A920	BringToFront	64K	Window Manager
$A90A	CalcVBehind	64K	Window Manager
$A909	CalcVis	64K	Window Manager
$A911	CheckUpdate	64K	Window Manager
$A90B	ClipAbove	64K	Window Manager
$A92D	CloseWindow	64K	Window Manager
$A914	DisposeWindow	64K	Window Manager
$A905	DragGrayRgn	64K	Window Manager
$A926	DragTheRgn	64K	Window Manager
$A925	DragWindow	64K	Window Manager
$A904	DrawGrowIcon	64K	Window Manager
$A90F	DrawNew	64K	Window Manager
$A923	EndUpDate	64K	Window Manager
$A92C	FindWindow	64K	Window Manager
$A924	FrontWindow	64K	Window Manager
$A9BD	GetNewWindow	64K	Window Manager
$A92F	GetWindowPic	64K	Window Manager
$A910	GetWMgrPort	64K	Window Manager
$A917	GetWRefCon	64K	Window Manager
$A919	GetWTitle	64K	Window Manager
$A92B	GrowWindow	64K	Window Manager
$A916	HideWindow	64K	Window Manager
$A91C	HiliteWindow	64K	Window Manager
$A912	InitWindows	64K	Window Manager
$A928	InvalRect	64K	Window Manager
$A927	InvalRgn	64K	Window Manager
$A91B	MoveWindow	64K	Window Manager
$A913	NewWindow	64K	Window Manager
$A90D	PaintBehind	64K	Window Manager
$A90C	PaintOne	64K	Window Manager
$A94E	PinRect	64K	Window Manager
$A90E	SaveOld	64K	Window Manager
$A91F	SelectWindow	64K	Window Manager
$A921	SendBehind	64K	Window Manager
$A92E	SetWindowPic	64K	Window Manager
$A918	SetWRefCon	64K	Window Manager
$A91A	SetWTitle	64K	Window Manager
$A908	ShowHide	64K	Window Manager
$A915	ShowWindow	64K	Window Manager
$A91D	SizeWindow	64K	Window Manager
$A83B	TrackBox	128K	Window Manager
$A91E	TrackGoAway	64K	Window Manager
$A92A	ValidRect	64K	Window Manager
$A929	ValidRgn	64K	Window Manager
$A83A	ZoomWindow	128K	Window Manager
$AA42	GetAuxWin	256K	Window Manager (color)
$AA48	GetCWMgrPort	256K	Window Manager (color)
$AA46	GetNewCWindow	256K	Window Manager (color)
$AA45	NewCWindow	256K	Window Manager (color)
$AA47	SetDeskCPat	256K	Window Manager (color)
$AA41	SetWinColor	256K	Window Manager (color)

(continued)

Table 3.005. Continued

Notes:

•* First ROM that includes trap.

•** For internal use only; not used by applications.

•*** Dispatched based upon additional information:

0	FSControl
1	PBOpenWD
2	PBCloseWD
5	PBCatMove
6	PBDirCreate
7	PBGetWDInfo
8	PBGetFCBInfo
9	PBGetCatInfo
10	PBSetCatInfo
11	PBSetVolInfo
16	PBLockRng
17	PBUnlockRng
$30	GetVolParms
$31	GetLogInInfo
$32	GetDirAccess
$33	SetDirAccess
$34	MapID
$35	MapName
$36	CopyFile
$37	MoveRename
$38	OpenDeny
$39	OpenRFDeny

•**** Dispatched based upon additional information:

1	ShutDwnPower
2	ShutDwnStart
3	ShutDwnInstall
4	ShutDwnRemove

•† Dispatched based upon additional information:

0	TEStylPaste
1	TESetStyle
2	TEReplaceStyle
3	TEGetStyle
4	GetStylHandle
5	SetStylHandle
6	GetStylScrap
7	TEStylInsert
8	TEGetPoint
9	TEGetHeight

Source: Inside Macintosh, pages III-215 to III-226, V-603 to V-605, X-Ref-63 to X-Ref-72

See Also: 3.008. Interface Toolbox Managers
3.156. Operating System Calls
And other tables for various Toolbox and OS routines

3.006 System Traps and Routines That May Move or Purge Memory

Trap Number	*Name*	*Moves Memory*
$AA94	ActivatePalette	Yes
$A07C	ADBOp	
$A07B	ADBReInit	Yes
$AA3B	AddComp	Yes
$A04E	AddDrive	
$A87E	AddPt	
$A9AC	AddReference	
$A94D	AddResMenu	Yes
$A9AB	AddResource	
$AA3A	AddSearch	Yes
$A985	Alert	Yes
$A010	Allocate	
$A20F	AllocConfig	
$AA1D	AllocCursor	Yes
$A8C4	AngleFromSlope	
$AA99	AnimateEntry	
$AA9A	AnimatePalette	
$A933	AppendMenu	Yes
AppleTalk*	ATPAddRsp	Yes
AppleTalk*	ATPCloseSocket	Yes
AppleTalk*	ATPGetRequest	Yes
AppleTalk*	ATPLoad	Yes
AppleTalk*	ATPOpenSocket	Yes
AppleTalk*	ATPReqCancel	Yes
AppleTalk*	ATPRequest	Yes
AppleTalk*	ATPResponse	Yes
AppleTalk*	ATPRspCancel	Yes
AppleTalk*	ATPSndRequest	Yes
AppleTalk*	ATPSndRsp	Yes
AppleTalk*	ATPUnload	Yes
$A071	AttachVBL	
$A863	BackColor	Yes
$A87C	BackPat	Yes
$AA0B	BackPixPat	Yes
$A922	BeginUpdate	Yes
$A85F	BitClr	
$A85A	BitNot	
$A85B	BitOr	
$A85E	BitSet	
$A85C	BitShift	
$A85D	BitTst	
$A859	BitXOr	
$A02E	BlockMove	
$A920	BringToFront	Yes
$A974	Button	Yes
$A838	CalcMask	
$AA4F	CalcMask	
$A948	CalcMenuSize	Yes
$A909	CalcVis	Yes
$A90A	CalcVisBehind	Yes
$A988	CautionAlert	Yes
$A9F3	Chain	Yes
$A9AA	ChangedResource	Yes
Script Manager*	Char2Pixel	Yes
$AA23	CharExtra	
$A88D	CharWidth	Yes
$A945	CheckItem	Yes
$A911	CheckUpdate	Yes
$A934	ClearMenuBar	
$A90B	ClipAbove	Yes

(continued)

Table 3.006. Continued

Trap Number	Name	Moves Memory
$A87B	ClipRect	Yes
$A87D	CloseCPort	
$A9B7	CloseDeskAcc	
$A982	CloseDialog	Yes
$A8F4	ClosePicture	Yes
$A8CC	ClosePoly	Yes
$A87D	ClosePort	
$A99A	CloseResFile	Yes
$A8DB	CloseRgn	Yes
$A92D	CloseWindow	Yes
ColorPicker*	CMY2RGB	Yes
$AA33	Color2Index	Yes
$A864	ColorBit	
$A04C	CompactMem	Yes
$A004	Control	Yes
$A8EC	CopyBits	Yes
$A817	CopyMask	Yes
$AA05	CopyPixMap	
$AA09	CopyPixPat	
$A8DC	CopyRgn	Yes
$A989	CouldAlert	Yes
$A979	CouldDialog	Yes
$A80D	Count1Resources	
$A81C	Count1Types	
$A077	CountADBs	
$A950	CountMItems	
$A99C	CountResources	
$A99E	CountTypes	
$A008	Create	
$A9B1	CreateResFile	Yes
$AA9F	CTab2Palette	Yes
$A994	CurResFile	
$A9C7	Date2Secs	
AppleTalk*	DDPCloseSocket	Yes
AppleTalk*	DDPOpenSocket	Yes
AppleTalk*	DDPRdCancel	Yes
AppleTalk*	DDPRead	Yes
AppleTalk*	DDPWrite	Yes
$A9FF	Debugger	
$ABFF	DebugStr	
$A03B	Delay	
$AA4D	DelComp	Yes
$A009	Delete	
$A936	DeleteMenu	
$AA60	DelMCEntries	Yes
$A952	DelMenuItem	Yes
$AA4C	DelSearch	Yes
$A94F	DeltaPoint	
$A96E	Dequeue	
$A992	DetachResource	
$A980	DialogSelect	Yes
$A8E6	DiffRgn	Yes
$A93A	DisableItem	
Disk Driver*	DiskEject	Yes
$AA63	DispMCInfo	Yes
$AA26	DisposCCursor	
$AA25	DisposCIcon	
$AA24	DisposCTable	Yes
$A983	DisposDialog	Yes
$A955	DisposeControl	Yes

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A932	DisposeMenu	Yes
$A8D9	DisposeRgn	Yes
$A914	DisposeWindow	Yes
$AA30	DisposGDevice	Yes
$A023	DisposHandle	Yes
$AA93	DisposPalette	Yes
$AA04	DisposPixMap	Yes
$AA08	DisposPixPat	Yes
$A01F	DisposPtr	Yes
TextEdit*	DlgCntl	Yes
TextEdit*	DlgCopy	Yes
TextEdit*	DlgDelete	Yes
TextEdit*	DlgPaste	
$A072	DoVBLTask	
$A967	DragControl	Yes
$A905	DragGrayRgn	Yes
$A926	DragTheRgn	
$A925	DragWindow	Yes
$A96D	Draw1Control	Yes
$A883	DrawChar	Yes
$A969	DrawControls	
$A981	DrawDialog	Yes
$A904	DrawGrowIcon	Yes
TextEdit*	DrawJust	Yes
$A937	DrawMenuBar	Yes
$A90F	DrawNew	Yes
$A8F6	DrawPicture	Yes
$A884	DrawString	Yes
$A885	DrawText	Yes
Disk Driver*	DriveStatus	Yes
$A03D	DrvrInstall	Yes
$A03E	DrvrRemove	Yes
$A017	Eject	Yes
$A02B	EmptyHandle	Yes
$A8AE	EmptyRect	
$A8E2	EmptyRgn	
$A939	EnableItem	
$A923	EndUpDate	Yes
$A96F	Enqueue	
$A881	EqualPt	
$A8A6	EqualRect	
$A8E3	EqualRgn	
$A03C	EqualString	
$A8C0	EraseArc	Yes
$A8B9	EraseOval	Yes
$A8C8	ErasePoly	Yes
$A8A3	EraseRect	Yes
$A8D4	EraseRgn	Yes
$A8B2	EraseRoundRect	Yes
$A98C	ErrorSound	
$A971	EventAvail	Yes
$A9F4	ExitToShell	Yes
$A8C2	FillArc	Yes
$AA11	FillCArc	Yes
$AA0F	FillCOval	Yes
$AA13	FillCPoly	Yes
$AA0E	FillCRect	Yes
$AA12	FillCRgn	Yes
$AA10	FillCRoundRect	Yes
$A8BB	FillOval	Yes

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A8CA	FillPoly	Yes
$A8A5	FillRect	Yes
$A8D6	FillRgn	Yes
$A8B4	FillRoundRect	Yes
$A96C	FindControl	Yes
$A984	FindDItem	Yes
$A92C	FindWindow	
TextEdit*	FindWord	Yes
ColorPicker*	Fix2SmallFract	Yes
$A818	FixATan2	
$A868	FixMul	
$A869	FixRatio	
$A86D	FixRound	
$A94C	FlashMenuBar	Yes
$A032	FlushEvents	
$A045	FlushFile	
$A013	FlushVol	Yes
$A901	FMSwapFont	Yes
Script Manager*	Font2Script	Yes
$A835	FontMetrics	Yes
Script Manager*	FontScript	Yes
$A862	ForeColor	Yes
$A8BE	FrameArc	Yes
$A8B7	FrameOval	Yes
$A8C6	FramePoly	Yes
$A8A1	FrameRect	Yes
$A8D2	FrameRgn	Yes
$A8B0	FrameRoundRect	Yes
$A98A	FreeAlert	Yes
$A97A	FreeDialog	Yes
$A01C	FreeMem	Yes
$A924	FrontWindow	
$A80E	Get1IndResource	Yes
$A80F	Get1IndType	Yes
$A820	Get1NamedResource	Yes
$A81F	Get1Resource	Yes
$A079	GetADBInfo	
$A9F5	GetAppParms	
$AA44	GetAuxCtl	Yes
$AA42	GetAuxWin	
$AA1A	GetBackColor	
$AA1B	GetCCursor	Yes
$AA1E	GetCIcon	Yes
$A87A	GetClip	Yes
ColorPicker*	GetColor	Yes
$AA17	GetCPixel	
$A95A	GetCRefCon	
$AA18	GetCTable	Yes
$A95E	GetCTitle	
$A96A	GetCtlAction	
$A962	GetCtlMax	
$A961	GetCtlMin	
$A960	GetCtlValue	
$AA28	GetCTSeed	
$A9B9	GetCursor	Yes
$A809	GetCVariant	
$AA48	GetCWMgrPort	
Disk Driver*	GetDCtlEntry	Yes
$A07D	GetDefaultStartup	
$AA29	GetDeviceList	

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A98D	GetDItem	Yes
$AA9B	GetEntryColor	
$AA9D	GetEntryUsage	
$A011	GetEOF	
$A00C	GetFileInfo	
$A900	GetFNum	Yes
$A88B	GetFontInfo	Yes
$A8FF	GetFontName	Yes
$AA19	GetForeColor	
$A018	GetFPos	
$AA32	GetGDevice	
Window Manager*	GetGrayRegion	Yes
$A025	GetHandleSize	
$A9BB	GetIcon	Yes
$A078	GetIndADB	
Toolbox Utilities*	GetIndPattern	
$A99D	GetIndResource	Yes
Toolbox Utilities*	GetIndString	
$A99F	GetIndType	
$A946	GetItem	
$A84E	GetItemCmd	
$A93F	GetItemIcon	
$A943	GetItemMark	
$A941	GetItemStyle	
$A976	GetKeys	Yes
$A990	GetIText	
$AA2A	GetMainDevice	
$A836	GetMaskTable	
$AA27	GetMaxDevice	
$AA64	GetMCEntry	
$AA61	GetMCInfo	Yes
$A9BF	GetMenu	Yes
$A93B	GetMenuBar	Yes
$A949	GetMHandle	
$A972	GetMouse	Yes
$A9A1	GetNamedResource	Yes
$A9BE	GetNewControl	Yes
$AA46	GetNewCWindow	Yes
$A97C	GetNewDialog	Yes
$A9C0	GetNewMBar	Yes
$AA92	GetNewPalette	Yes
$A9BD	GetNewWindow	Yes
$AA2B	GetNextDevice	
$A970	GetNextEvent	Yes
$A084	GetOSDefault	
$A031	GetOSEvent	
$AA96	GetPalette	
$A9B8	GetPattern	Yes
$A89A	GetPen	
$A898	GetPenState	
$A9BC	GetPicture	Yes
$A865	GetPixel	
$AA0C	GetPixPat	Yes
$A874	GetPort	
$A021	GetPtrSize	
$A9A6	GetResAttrs	
$A9F6	GetResFileAttrs	
$A9A8	GetResInfo	
$A9A0	GetResource	Yes
$A9FD	GetScrap	Yes

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A9BA	GetString	Yes
TextEdit*	GetStylHandle	Yes
TextEdit*	GetStylScrap	Yes
$AA37	GetSubTable	Yes
$A146	GetTrapAddress	
$A080	GetVideoDefault	
$A014	GetVol	
$A007	GetVolInfo	
$A92F	GetWindowPic	
$A910	GetWMgrPort	
$A917	GetWRefCon	
$A919	GetWTitle	
$A11A	GetZone	
$A871	GlobalToLocal	
$A872	GrafDevice	
$A92B	GrowWindow	Yes
$A9E4	HandAndHand	Yes
$A126	HandleZone	
$A9E1	HandToHand	Yes
$A068	HClrRBit	
$A208	HCreate	
$A209	HDelete	
$A260	HFSDispatch	
$A20B	HGetFileInfo	
$A069	HGetState	
$A214	HGetVol	
$A207	HGetWDInfo	
$A958	HideControl	Yes
$A852	HideCursor	
$A827	HideDItem	Yes
$A896	HidePen	
$A916	HideWindow	Yes
$AA22	HiliteColor	
$A95D	HiliteControl	Yes
$A938	HiliteMenu	Yes
TextEdit*	HiliteText	Yes
$A91C	HiliteWindow	Yes
$A86B	HiWord	
$A029	HLock	
$A04A	HNoPurge	
$A9A4	HomeResFile	
$A200	HOpen	
$A20A	HOpenRF	
$A049	HPurge	
$A20C	HRename	
$A242	HRstFLock	
$A20E	HSetFileInfo	
$A241	HSetFLock	
$A067	HSetRBit	
$A06A	HSetState	
$A215	HSetVol	
ColorPicker*	HSL2RGB	Yes
ColorPicker*	HSV2RGB	Yes
$A02A	HUnlock	
$AA34	Index2Color	
$A9F9	InfoScrap	
$A9E6	InitAllPacks	Yes
$A02C	InitApplZone	Yes
$AA01	InitCPort	Yes
$A850	InitCursor	

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A97B	InitDialogs	
$A8FE	InitFonts	Yes
$A06C	InitFS	
$AA2E	InitGDevice	Yes
$A86E	InitGraf	Yes
$A930	InitMenus	Yes
$A9E5	InitPack	Yes
$AA90	InitPalettes	Yes
$A86D	InitPort	Yes
Slot Manager*	InitPRAMRecs	Yes
$A808	InitProcMenu	Yes
$A016	InitQueue	
$A995	InitResources	Yes
Slot Manager*	InitSDeclMgr	Yes
Slot Manager*	InitSRsrcTable	Yes
$A03F	InitUtil	
$A912	InitWindows	Yes
$A019	InitZone	Yes
$A935	InsertMenu	Yes
$A951	InsertResMenu	Yes
$A8A9	InsetRect	
$A8E1	InsetRgn	Yes
$A826	InsMenuItem	Yes
$A058	InsTime	
Script Manager*	Int1Script	Yes
$A07F	InternalWait	
$A928	InvalRect	Yes
$A927	InvalRgn	Yes
$A8C1	InvertArc	Yes
$AA35	InvertColor	
$A8BA	InvertOval	Yes
$A8C9	InvertPoly	Yes
$A8A4	InvertRect	Yes
$A8D5	InvertRgn	Yes
$A8B3	InvertRoundRect	Yes
$A97F	IsDialogEvent	
TextEdit*	KeyScript	Yes
$A9C3	KeyTrans	
$A956	KillControls	Yes
$A006	KillIO	
$A8F5	KillPicture	Yes
$A8CD	KillPoly	Yes
AppleTalk*	LAPCloseProtocol	Yes
AppleTalk*	LAPOpenProtocol	Yes
AppleTalk*	LAPRdCancel	Yes
AppleTalk*	LAPRead	Yes
AppleTalk*	LAPWrite	Yes
$A9F2	Launch	Yes
$A892	Line	Yes
$A891	LineTo	Yes
$A9A2	LoadResource	Yes
$A9FB	LoadScrap	Yes
$A9F0	LoadSeg	Yes
$A870	LocalToGlobal	
$A867	LongMul	
$A86B	LoWord	
$AA39	MakeITable	Yes
$AA0D	MakeRGBPat	
$A8FC	MapPoly	
$A8F9	MapPt	

(continued)

Table 3.006. Continued

Trap Number	Name	Moves Memory
$A8FA	MapRect	
$A8FB	MapRgn	Yes
$A063	MaxApplZone	
$A061	MaxBlock	
$A11D	MaxMem	
$A821	MaxSizeRsrc	
TextEdit*	MeasureJust	Yes
$A837	MeasureText	Yes
$AA66	MenuChoice	
$A93E	MenuKey	Yes
$A93D	MenuSelect	Yes
$A991	ModalDialog	Yes
$A036	MoreMasters	Yes
$A00F	MountVol	
$A894	Move	
$A959	MoveControl	Yes
$A064	MoveHHi	Yes
$A877	MovePortTo	
$A893	MoveTo	
$A91B	MoveWindow	Yes
AppleTalk*	MPPClose	Yes
AppleTalk*	MPPOpen	Yes
$A9E0	Munger	Yes
$AA4B	NewCDialog	Yes
$A954	NewControl	Yes
$AA45	NewCWindow	Yes
$A97D	NewDialog	Yes
$A166	NewEmptyHandle	Yes
$AA2F	NewGDevice	Yes
$A122	NewHandle	Yes
$A931	NewMenu	Yes
$AA91	NewPalette	Yes
$AA03	NewPixMap	Yes
$AA07	NewPixPat	Yes
Color QuickDraw*	NewPort	Yes
$A11E	NewPtr	Yes
$A8D8	NewRgn	Yes
$A906	NewString	Yes
$A913	NewWindow	Yes
$A987	NoteAlert	Yes
$A856	ObscureCursor	
$A035	Offline	
$A8CE	OffsetPoly	
$A8A8	OffsetRect	
$A8E0	OffsetRgn	
$AA21	OpColor	
$A000	Open	
Color QuickDraw*	OpenCPicture	Yes
$AA00	OpenCPort	Yes
$A9B6	OpenDeskAcc	Yes
$A8F3	OpenPicture	Yes
Color QuickDraw*	OpenPixMap	Yes
$A8CB	OpenPoly	Yes
$A86F	OpenPort	Yes
$A997	OpenResFile	Yes
$A00A	OpenRF	Yes
$A9C4	OpenRFPerm	Yes
$A8DA	OpenRgn	Yes
$A030	OSEventAvail	
$A9E7	Pack0	

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A9E8	Pack1	
$A82C	Pack10	
$A82D	Pack11	
$A82E	Pack12 (ColorPicker)	
$A82F	Pack13	
$A830	Pack14	
$A831	Pack15	
$A9E9	Pack2	Yes **
$A9EA	Pack3 (StdFile)	Yes ***
$A9EB	Pack4 (Floating Point)	
$A9EC	Pack5 (Transcendental Functions)	
$A9ED	Pack6 (IntUtils)	Yes ****
$A9EE	Pack7 (Binary-Decimal Conversion)	Yes †
$A816	Pack8	
$A82B	Pack9	
$A8CF	PackBits	
$A8BF	PaintArc	Yes
$A90D	PaintBehind	Yes
$A90C	PaintOne	Yes
$A8B8	PaintOval	Yes
$A8C7	PaintPoly	Yes
$A8A2	PaintRect	Yes
$A8D3	PaintRgn	Yes
$A8B1	PaintRoundRect	Yes
$AAA0	Palette2CTab	Yes
$A98B	ParamText	Yes
$A001	PBClose	
Disk Driver*	PBControl	Yes
Disk Driver*	PBEject	Yes
Disk Driver*	PBFlushVol	Yes
Disk Driver*	PBMountVol	Yes
Disk Driver*	PBOffLine	Yes
Disk Driver*	PBOpen	Yes
Disk Driver*	PBOpenRF	Yes
Disk Driver*	PBStatus	Yes
$A89C	PenMode	
$A89E	PenNormal	Yes
$A89D	PenPat	Yes
$AA0A	PenPixPat	Yes
$A89B	PenSize	
$A8F2	PicComment	Yes
$A94E	PinRect	
Script Manager*	Pixel2Char	Yes
$AA1F	PlotCIcon	Yes
$A94B	PlotIcon	Yes
$AA98	PmBackColor	Yes
$AA97	PmForeColor	Yes
$A80B	PopUpMenuSelect	
$A876	PortSize	
$A02F	PostEvent	
$A12F	PPostEvent	
Printing Manager*	PrClose	Yes
Printing Manager*	PrCloseDoc	Yes
Printing Manager*	PrClosePage	Yes
Printing Manager*	PrCtlCall	Yes
Printing Manager*	PrDrvrClose	Yes
Printing Manager*	PrDrvrDCE	Yes
Printing Manager*	PrDrvrOpen	Yes
Printing Manager*	PrError	Yes
Printing Manager*	PrGeneral	Yes

(continued)

Table 3.006. Continued

Trap Number	Name	Moves Memory
$A8FD	PrGlue	
$A05A	PrimeTime	
Printing Manager*	PrintDefault	Yes
Printing Manager*	PrJobDialog	Yes
Printing Manager*	PrJobMerge	Yes
Printing Manager*	PrOpen	Yes
Printing Manager*	PrOpenDoc	Yes
Printing Manager*	PrOpenPage	Yes
$AA3D	ProtectEntry	
Printing Manager*	PrPicFile	Yes
Printing Manager*	PrSetError	Yes
Printing Manager*	PrStlDialog	Yes
Printing Manager*	PrValidate	Yes
$A8AC	Pt2Rect	
$A8AD	PtInRect	
$A8E8	PtInRgn	
$A9EF	PtrAndHand	Yes
$A9E3	PtrToHand	Yes
$A9E2	PtrToXHand	Yes
$A048	PtrZone	
$A148	PtrZone	
$A8C3	PtToAngle	
$A04D	PurgeMem	Yes
$A062	PurgeSpace	
$A9CA	PutIcon	
$A9FE	PutScrap	Yes
$AA40	QDError	
$A861	Random	
$A04F	RDrvrInstall	
$A002	Read	
$A039	ReadDateTime	
$A037	ReadParam	
$A051	ReadXPRam	
$AA36	RealColor	Yes
$A902	RealFont	Yes
$A027	ReallocHandle	Yes
$A128	RecoverHandle	
$A8E9	RectInRgn	
$A8DF	RectRgn	Yes
$A9A3	ReleaseResource	Yes
$A050	RelString	
$A00B	Rename	
$A9AF	ResError	
$AA3E	ReserveEntry	
$A040	ResrvMem	Yes
Shutdown*	Restart	Yes
$AA4A	RestoreEntries	
ColorPicker*	RGB2CMY	Yes
ColorPicker*	RGB2HSU	Yes
ColorPicker*	RGB2S2	Yes
$AA15	RGBBackColor	Yes
$AA14	RGBForeColor	Yes
$A80C	RGetResource	Yes
$A9AE	RmveReference	
$A9AD	RmveResource	Yes
$A059	RmvTime	
$A9C5	RsrcMapEntry	
$A996	RsrcZoneInit	Yes
$A042	RstFilLock	
$AA49	SaveEntries	

(continued)

Table 3.006. Continued

Trap Number	*Name*	*Moves Memory*
$A90E	SaveOld	Yes
$A8F8	ScalePt	
$A8B5	ScriptUtil	
$A833	ScrnBitMap	
$A8EF	ScrollRect	Yes
$A9C6	Secs2Date	
$A8AA	SectRect	
$A8E4	SectRgn	Yes
$AA50	SeedCFill	
$A839	SeedFill	
$A91F	SelectWindow	Yes
$A97E	SellText	Yes
$A921	SendBehind	Yes
Serial*	SerClrBrk	Yes
Serial*	SerGetBrk	Yes
Serial*	SerHShake	Yes
Serial*	SerReset	Yes
Serial*	SerSetBrk	Yes
Serial*	SerSetBuf	Yes
Serial*	SerStatus	Yes
$A07A	SetADBInfo	
$A057	SetApplBase	Yes
$A02D	SetAppLimit	
$AA1C	SetCCursor	Yes
$AA3C	SetClientID	
$A879	SetClip	Yes
$AA16	SetCPixel	Yes
$A95B	SetCRefCon	
$A95F	SetCTitle	Yes
$A96B	SetCtlAction	
$AA43	SetCtlColor	Yes
$A965	SetCtlMax	Yes
$A964	SetCtlMin	Yes
$A963	SetCtlValue	Yes
$A851	SetCursor	
$A03A	SetDateTime	
$A07E	SetDefaultStartup	
$AA47	SetDeskCPat	Yes
$AA2D	SetDeviceAttribute	
$A98E	SetDItem	Yes
$A8DD	SetEmptyRgn	Yes
$AA3F	SetEntries	
$AA9C	SetEntryColor	
$AA9E	SetEntryUsage	
$A012	SetEOF	
$A00D	SetFileInfo	
$A041	SetFilLock	
$A043	SetFilType	
$A903	SetFontLock	Yes
$A044	SetFPos	
$A814	SetFractEnable	
$A834	SetFScaleDisable	
$AA31	SetGDevice	
$A04B	SetGrowZone	
$A024	SetHandleSize	Yes
$A947	SetItem	Yes
$A84F	SetItemCmd	
$A940	SetItemIcon	Yes
$A944	SetItemMark	Yes
$A942	SetItemStyle	Yes

(continued)

Table 3.006. Continued

Trap Number	Name	Moves Memory
$A98F	SetIText	Yes
$AA65	SetMCEntries	
$AA62	SetMCInfo	Yes
$A93C	SetMenuBar	
$A94A	SetMenuFlash	
$A878	SetOrigin	
$A083	SetOSDefault	
$AA95	SetPalette	
$A899	SetPenState	
$A873	SetPort	
$A875	SetPortBits	
$AA06	SetPortPix	
$A880	SetPt	
$A020	SetPtrSize	Yes
$A8A7	SetRect	
$A8DE	SetRectRgn	Yes
$A9A7	SetResAttrs	
$A9F7	SetResFileAttrs	
$A9A9	SetResInfo	Yes
$A99B	SetResLoad	
$A993	SetResPurge	
$AA4E	SetStdCProcs	
$A8EA	SetStdProcs	
$A907	SetString	Yes
$A047	SetTrapAddress	
$A081	SetVideoDefault	
$A015	SetVol	
$AA41	SetWinColor	Yes
$A92E	SetWindowPic	
$A918	SetWRefCon	
$A91A	SetWTitle	Yes
$A01B	SetZone	
$A855	ShieldCursor	
$A957	ShowControl	Yes
$A853	ShowCursor	
$A828	ShowDItem	Yes
$A908	ShowHide	Yes
$A897	ShowPen	
$A915	ShowWindow	Yes
$A895	ShutDown Dispatch	Yes ¥
$A075	SIntInstall	
$A076	SIntRemove	
$A95C	SizeControl	Yes
$A9A5	SizeResource	
$A91D	SizeWindow	Yes
$A8BC	SlopeFromAngle	
$A06E	SlotManager	Yes §
$A06F	SlotVInstall	
$A070	SlotVRemove	
ColorPicker*	SmallFract2Fix	Yes
$A802	SndAddModifier	Yes
$A806	SndControl	
Sound*	SndDisposeChannel	Yes
$A803	SndDoCommand	
$A804	SndDoImmediate	
$A807	SndNewChannel	Yes
$A805	SndPlay	
$A88E	SpaceExtra	
$A065	StackSpace	
Sound*	StartSound	Yes

(continued)

Table 3.006. Continued

Trap Number	Name	Moves Memory
$A005	Status	Yes
$A8BD	StdArc	Yes
$A8EB	StdBits	Yes
$A8F1	StdComment	Yes
$A8EE	StdGetPic	
$A890	StdLine	Yes
$ABF8	StdOpcodeProc	
$A8B6	StdOval	Yes
$A8C5	StdPoly	Yes
$A8F0	StdPutPic	Yes
$A8A0	StdRect	Yes
$A8D1	StdRgn	Yes
$A8AF	StdRRect	Yes
$A882	StdText	Yes
$A8ED	StdTxMeas	Yes
$A973	StillDown	Yes
$A986	StopAlert	Yes
Sound*	StopSound	Yes
$A88C	StringWidth	Yes
$A055	StripAddress	
$A866	StuffHex	
$A87F	SubPt	
$A05D	SwapMMUMode	
$A9C8	SysBeep	Yes
$A090	SysEnvirons	
$A9C9	SysError	Yes
$A9B3	SystemClick	Yes
$A9C2	SystemEdit	Yes
$A9B2	SystemEvent	
$A9B5	SystemMenu	Yes
$A9B4	SystemTask	
$A9D8	TEActivate	Yes
$A813	TEAutoView	Yes
$A9D0	TECalText	Yes
$A9D4	TEClick	Yes
$A9D5	TECopy	Yes
$A9D6	TECut	Yes
$A9D9	TEDeactivate	Yes
$A9D7	TEDelete	Yes
$A83D	TEDispatch	Yes ‡
$A9CD	TEDispose	Yes
TextEdit*	TEFromScrap	Yes
$A83C	TEGetOffset	Yes
$A9CB	TEGetText	Yes
$A9DA	TEIdle	Yes
$A9CC	TEInit	Yes
$A9DE	TEInsert	Yes
$A9DC	TEKey	Yes
$A9D2	TENew	Yes
$A9DB	TEPaste	Yes
$A812	TEPinScroll	Yes
$A9DD	TEScroll	Yes
$A811	TESelView	Yes
$A9DF	TESetJust	Yes
$A9D1	TESetSelect	Yes
$A9CF	TESetText	Yes
$A966	TestControl	Yes
$AA2C	TestDeviceAttribute	
$A83E	TEStyleNew	
$A9D3	TEUpdate	Yes

(continued)

Table 3.006. Continued

Trap Number	Name	Moves Memory
$A9CE	TextBox	Yes
$A888	TextFace	
$A887	TextFont	
$A889	TextMode	
$A88A	TextSize	
$A886	TextWidth	Yes
$A975	TickCount	Yes
$A83B	TrackBox	Yes
$A968	TrackControl	Yes
$A91E	TrackGoAway	Yes
Script Manager*	Transliterate	Yes
$A8AB	UnionRect	
$A8E5	UnionRgn	Yes
$A810	Unique1ID	
$A9C1	UniqueID	
$A9FA	UnloadScrap	Yes
$A9F1	UnloadSeg	Yes
$A00E	UnmountVol	
$A8D0	UnpackBits	
$A999	UpdateResFile	
$A953	UpdtControl	Yes
$A978	UpdtDialog	Yes
$A054	UprString	
$A998	UseResFile	
$A92A	ValidRect	Yes
$A929	ValidRgn	Yes
$A033	VInstall	
$A034	VRemove	
$A977	WaitMouseUp	Yes
$A003	Write	
$A038	WriteParam	
$A9B0	WriteResource	
$A052	WriteXPRam	
$A819	XMunger	
$A8E7	XorRgn	Yes
$A9FC	ZeroScrap	Yes
$A83A	ZoomWindow	Yes

Notes:

- * This routine is not a trap but either a dispatched routine or a routine controlled by the indicated manager.
- ** DIBadMount, DIFormat, DILoad, DIUnload, DIVerify, DIZero.
- *** SFGetFile, SFPGetFile, SFPPutFile, SFPutFile.
- **** IUCompString, IUDatePString, IUDateString, IUEqualString, IUGetIntl, IUMagIDString, IUMagString, IUMetric, IUSetIntl, IUTimePString, IUTimeString.
- † NumToString, StringToNum, PStr2Dec, Dec2Str, CStr2Dec.
- ¥ ShutDwnInstall, ShutDwnRemove.
- § sExec, sGetBlock, sGetcString, sGetDriver, sPrimaryInit.
- ‡ TEGetHeight, TEGetOffset, TEGetPoint, TEGetStyle, TEReplaceStyle, TESetStyle, TEStylInsert, TE StyleNew, TEStylPaste.

Source: Inside Macintosh, pages III-215 to III-226, V-603 to V-605, X-Ref-63 to X-Ref-72

See Also:

3.008. Interface Toolbox Managers
3.156. Operating System Calls
And other tables for various Toolbox and OS routines

3.007. Global Variables

Name	*Location **	*Size*	*Description/Comments*
ABusVars	$2D8		Pointer to AppleTalk variables
ACount	$A9A	Word	Stage number (0–3) of last alert
ANumber	$A98	Word	Resource ID of last alert
ApFontID	$984	Word	Number of application font
ApplLimit	$130		Application heap limit
ApplScratch	$A78	12 bytes	Application scratch area
ApplZone	$2AA		Address of application heap
AppParmHandle	$AEC		Handle to Finder info
AtMenuBottom	$A0C	Word	Menu scrolling flag
AuxWinHead	$CD0	Long	Auxiliary window list header
BootDrive	$210	Word	Working directory reference number for system startup volume
BufPtr	$10C		Address of end of jump table
BufTgDate	$304	Long	Date and time of last modification file tag
BufTgFBkNum	$302	Word	Logical block number file tag
BufTgFFlg	$300	Word	Flags file tag (bit 1 = 1 if resource fork)
BufTgFNum	$2FC	Long	File number file tag
CaretTime	$2F4	Long	Caret-blink interval (in ticks)
CPUFlag	$12F	Word	CPU in use flag
CrsrThresh	$8EC	Word	Mouse-scaling threshold
CurActivate	$A64		Pointer to window to receive activate event
CurApName	$910	32 bytes	Current application name (Pascal string)
CurApRefNum	$900	Word	Reference number of current application's resource file
CurDeactive	$A68		Pointer to window to receive deactivate event
CurDirStore	$398	Long	ID of directory last opened
CurJTOffset	$934	Word	Offset to jump table from A5
CurMap	$A5A	Word	Reference number of current resource file
CurPageOption	$936	Word	Sound/Screen buffer config passed to Chain/Launch
CurPitch	$280	Word	Count in square-wave synthesizer buffer
CurrentA5	$904		Address of boundary between application globals and parameters
CurStackBase	$908		Address of stack base (start of application globals)
DABeeper	$A9C		Address of current sound procedure
DAStrings	$AA0	16 bytes	Handles to ParamText strings
DefltStack	$322	Long	Default stack space allotment
DefVCBPtr	$352		Pointer to default VCB
DeskHook	$A6C		Address of painting desktop/clicks response procedure
DeskPattern	$A3C	8 bytes	Desktop pattern
DeviceList	$8A8		Handle to first element in device list
DlgFont	$AFA	Word	Font number used in dialogs and alerts
DoubleTime	$2F0	Long	Double-click interval (in ticks)
DragHook	$9F6		Address of drag control procedure
DragPattern	$A34	8 bytes	Dragged region's outline pattern
DrvQHdr	$308	10 bytes	Drive queue header
DSAlertRect	$3F8	8 bytes	Rectangle enclosing system error alert
DSAlertTab	$2BA		Pointer to system error alert table
DSErrCode	$AF0	Word	Current system error ID
DTQueue	$D92	10 bytes	Deferred task queue header
EventQueue	$A14A	10 bytes	Event queue header
ExtStsDT	$2BE	16 bytes	External status interrupt vector table
FCBSPtr	$34E		Pointer to FCB buffer
FinderName	$2E0	16 bytes	Finder name (Pascal string)
FractEnable	$BF4	Byte	Fractional widths flag (nonzero = enables)
FScaleDisable	$A63	Byte	Font scaling flag (nonzero to disable)
FSFCBLen	$3F6	Word	Size of a FCB (-1 on 64K ROM)
FSQHdr	$360	10 bytes	File I/O queue header
GhostWindow	$A84		Pointer to window never considered foremost
GrayRgn	$9EE		Handle to desktop region
GZRootHnd	$328		Handle to relocatable block not to be moved by GrowZone
HeapEnd	$114		Address of application heap zone end
HiliteMode	$938		Highlighting flag
HiliteRGB	$DA0		Default highlight color
IntlSpec	$BA0	Long	International installed flag (-1 = no)
JADBProc	$6B8		Pointer to ADBReInit pre/postprocessing routine
JDTInstall	$D9C		Jump vector for DTInstall routine
JFetch	$8F4		Jump vector for Fetch function
JIODone	$8FC		Jump vector for IODone function
Journal Flag	$8DE	Word	Journaling mode
JournalRef	$8E8	Word	Reference number of journaling device driver
JStash	$8F8		Jump vector for Stash function
JVBLTask	$D28		Jump vector for DoVBLTask routine
KbdLast	$218	Byte	ADB address of keyboard last used
KbdType	$21E	Byte	Type of keyboard last used

(continued)

Table 3.007. Continued

Name	Location *	Size	Description/Comments
KeyRepThresh	$190	Word	Auto-key rate
KeyThresh	$18E	Word	Auto-key threshold
LastFOND	$BC2		Handle to last family record used
Lo3Bytes	$31A		$00FFFFFF
Lvl1DT	$192	32 bytes	Level-1 secondary interrupt vector table
Lvl2DT	$1B2	32 bytes	Level-2 secondary interrupt vector table
MainDevice	$8A4		Handle to the current main device
MBarEnable	$A20	Word	Menu ID for DA
MBarHeight	$BAA	Word	Menu bar height
MBarHook	$A2C		Address of routine called before MenuSelect
MemErr	$220	Word	Current value of MemError
MemTop	$108		Address of RAM end
MenuCInfo	$D50		Header for menu color information table
MenuDisable	$B54		ID and item number for selected disabled item
MenuFlash	$A24	Word	Count of menu item blink duration
MenuHook	$A30		Address of routine called during MenuSelect
MenuList	$A1C		Handle to current menu list
MinStack	$31E	Long	Minimum stack space allotment
MinusOne	$A06		$FFFFFFFF
MMU32Bit	$CB2	Byte	Current address mode
OldContent	$9EA		Handle to saved content region
OldStructure	$9E6		Handle to saved structure region
OneOne	$A02		$00010001
PaintWhite	$9DC	Word	Paint window white before update flag
PortBUse	$291	Byte	Serial Port B Available flag
PrintErr	$944	Word	Result code from lasting Print Manager routine
QDColors	$8B0		Default QuickDraw colors
RAMBase	$2B2		Base address for Trap routines in RAM
ResErr	$A60	Word	Current value of ResError
ResErrProc	$AF2		Address of resource error procedure
ResLoad	$A5E	Word	Current SetResLoad state
ResumeProc	$A8C		Address of resume procedure
RndSeed	$156	Long	Random number seed
ROM85	$28E	Word	Version number of ROM
ROMBase	$2AE		Base address of ROM
ROMFont0	$980		Handle to record for system font
RomMapInsert	$B9E	Byte	Insert map to the ROM resources flag
SaveUpdate	$9DA	Word	Generate update events flag
SaveVisRgn	$9F2		Handle to saved visRgn
SCCRd	$1D8		SCC read (base address)
SCCWr	$1DC		SCC write (base address)
ScrapCount	$968	Word	Count changed by ZeroScrap
ScrapHandle	$964		Handle to desk scrap in memory
ScrapName	$96C		Pointer to scrap file name (Pascal string)
ScrapSize	$960	Long	Size of desk scrap (in bytes)
ScrapState	$96A	Word	Location of desk scrap
Scratch20	$1E4	20 bytes	Scratch area
Scratch8	$9FA	8 bytes	Scratch area
ScrDmpEnb	$2F8	Byte	Command-Shift-Number flag (0 = don't process)
ScrHRes	$104	Word	Horizontal pixels (per inch)
ScrnBase	$824		Address of main screen buffer
ScrVRes	$102	Word	Vertical pixels (per inch)
SdVolume	$260	Byte (LO 3bits)	Speaker volume
SEvtEnb	$15C	Byte	SystemEvent False flag (0 = return False)
SFSaveDisk	$214	Word	Negated volume reference number of Standard File Package
SoundBase	$266		Pointer to freeform synthesizer buffer
SoundLevel	$27F	Byte	Amplitude (in 740-byte buffer)
SoundPtr	$262		Pointer to four-tone record
SPAlarm	$200	Long	Alarm setting
SPATalkA	$1F9	Byte	AppleTalk node ID Hint (modem port)
SPATalkB	$1FA	Byte	AppleTalk node ID Hint (printer port)
SPClikCaret	$209	Byte	Double-click/caret-blink times
SPConfig	$1FB	Byte	Use types for serial ports
SPFont	$204	Word	Application font number, minus 1
SPKbd	$206	Byte	Auto-click threshold/rate
SPMisc2	$20B	Byte	Mouse scaling, system startup disk, menu blink values
SPPortA	$1FC	Word	Port configuration (modem)
SPPortB	$1FE	Word	Port configuration (printer)
SPPrint	$207	Byte	Printer connection
SPValid	$1F8	Byte	Validity status
SPVolCtl	$208	Byte	Speaker volume setting (in Parameter RAM)
SynListHandle	$D32		Handle to the synthetic font list

(continued)

Table 3.007. Continued

Name	*Location **	*Size*	*Description/Comments*
SysEvtMask	$144	Word	System event mask
SysFontFam	$BA6	Word	Number of font to use for system font (if non-zero)
SysFontSize	$BA8	Word	Size of system font (if non-zero)
SysMap	$A58	Word	Reference number of system resource file
SysMapHndl	$A54		Handle to map of system resource file
SysParam	$1F8	20 bytes	Copy of Parameter RAM
SysResName	$AD8	20 bytes	Name of system resource file (Pascal string)
SysZone	$2A6		Address of system heap
TEDoText	$A70		Address of TextEdit multipurpose routine
TERecal	$A74		Address of routine to recalculate TextEdit line starts
TEScrpHandle	$AB4		Handle to TextEdit scrap
TEScrpLength	$AB0	Long	Size of TextEdit scrap (in bytes)
TheGDevice	$CC8	Long	Handle to current active device
TheMenu	$A26	Word	ID of currently highlighted menu
TheZone	$118		Address of current heap zone
Ticks	$16A	Long	Number of ticks since system startup
Time	$20C	Long	Seconds since midnight, 1 January 1904
TimeDBRA	$D00	Word	DBRA instruction execution (times per millisecond)
TimeSCCDB	$D02	Word	SCC access rate (times per millisecond)
TimeSCSIDB	$DA6	Word	SCSI access rate (times per millisecond)
TmpResLoad	$B9F	Byte	Temporary SetResLoad state
ToExtFS	$3F2		Pointer to external file system
ToolScratch	$9CE	8 bytes	Scratch area
TopMapHndl	$A50		Handle to resource map of most recently opened resource file
TopMenuItem	$A0A		Top of scrollable menu (pixel value)
UTableBase	$11C		Base address of unit table
VBLQueue	$160	10 bytes	Vertical retrace queue header
VCBQHdr	$356	10 bytes	VCB queue header
VIA	$1DA		VIA base address
WidthListHand	$8E4		Handle to handle list of recently used width tables
WidthPtr	$B10		Pointer to global width table
WidthTabHandle	$B2A		Handle to global width table
WindowList	$9D6		Pointer to first window in list (0 = if not using windows)
WMgrPort	$9DE		Pointer to Window Manager port

Notes: * Addresses are not guaranteed by Apple and should be avoided if possible.

Source: Inside Macintosh, pages I-85, III-227 to III-231, IV-309, V-607, X-Ref-83 to X-Ref-88

3.008. Interface Toolbox Managers

Manager	*ROM Introduced **	*See Tables*
Resource Manager	64K	3.009 to 3.010
QuickDraw	64K	3.011 to 3.047
Color Manager	256K (Mac II)	3.048 to 3.051
Font Manager	64K	3.052 to 3.060
Script Manager	256K	3.061 to 3.066
Toolbox Event Manager	64K	3.067 to 3.072
Window Manager	64K	3.073 to 3.082
Control Manager	64K	3.083 to 3.091
Menu Manager	64K	3.092 to 3.101
TextEdit	64K	3.102 to 3.111
Dialog Manager	64K	3.112 to 3.120
Desk Manager	64K	3.121 to 3.124
Scrap Manager	64K	3.125 to 3.128
Toolbox Utilities	64K	3.129 to 3.133
Package Manager	64K	3.134 to 3.146
Binary-Decimal Conversion Package	64K	3.136
International Utilities Package	64K	3.137 to 3.141
Standard File Package	64K	3.142 to 3.146
Palette Manager	256K (Mac II)	3.147 to 3.150
List Manager	128K	3.151 to 3.155

Notes: * ROM where manager first appeared.

Source: Technical Introduction to the Macintosh Family, pages 16 to 21

See Also: 3.156. Operating System Calls

3.009. Resource Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
AddResource	Adds resource to current file	I	PROCEDURE AddResource (theData:Handle; theType:ResType; theID:INTEGER; name:Str255);
ChangedResource	Sets resource change attribute	I	PROCEDURE ChangedResource (theResource:Handle);
CloseResFile	Closes resource file	I	PROCEDURE CloseResFile (refNum:INTEGER);
Count1Resources	Counts resources in current resource file	IV	FUNCTION Count1Resources (theType:ResType) : INTEGER;
Count1Types	Counts types of current resource file	IV	FUNCTION Count1Types :INTEGER;
CountResources	Counts resources of a type in all open res files	I	FUNCTION CountResources (theType:ResType) : INTEGER;
CountTypes	Returns number of resource types in all open files	I	FUNCTION CountTypes : INTEGER;
CreateResFile	Creates new, empty resource file	I	PROCEDURE CreateResFile (fileName:Str255)
CurResFile	Returns reference number of current resource file	I	FUNCTION CurResFile : INTEGER;
DetachResource	Replaces handle of resource with NIL	I	PROCEDURE DetachResource (theResource:Handle);
Get1IndResource	Gets indexed resource in current resource file	IV	FUNCTION Get1IndResource (theType:ResType; index: INTEGER) : Handle;
Get1IndType	Gets index of resource in current resource file	IV	PROCEDURE Get1IndType (VAR theType: ResType; index:INTEGER);
Get1NamedResource	Gets named resource in current resource file	IV	FUNCTION Get1NamedResource (theType:ResType; name:Str255) : Handle;
Get1Resource	Gets resource in current resource file	IV	FUNCTION Get1Resource (theType:ResType; theID: INTEGER) : Handle;
GetIndResource	Gets handle to indexed resource	I	FUNCTION GetIndResource (theType:ResType; index:INTEGER) : Handle;
GetIndType	Gets handle to indexed resource type	I	PROCEDURE GetIndType (VAR theType: ResType; index:INTEGER);
GetNamedResource	Gets named resource in all open files	I	FUNCTION GetNamedResource (theType:ResType; name:Str255) : Handle;
GetResAttrs	Returns attributes for resource	I	FUNCTION GetResAttrs (theResource:Handle) : INTEGER;
GetResFileAttrs	Returns resource file attributes	I	FUNCTION GetResFileAttrs (refNum:INTEGER) : INTEGER;
GetResInfo	Returns information about resource	I	PROCEDURE GetResInfo (theResource:Handle; VAR theID:INTEGER; VAR theType:ResType; VAR name:Str255);
GetResource	Gets handle to resource	I	FUNCTION GetResource (theType:ResType; theID: INTEGER) : Handle;
HomeResFile	Returns reference number of res file containing resource	I	FUNCTION HomeResFile (theResource:Handle) : INTEGER;
InitResources	Called by system; not for application use	I	FUNCTION InitResources : INTEGER;
LoadResource	Reads resource into memory	I	PROCEDURE LoadResource (theResource:Handle);
MaxSizeRsrc	Determines size of resource from resource map	IV	FUNCTION MaxSizeRsrc (theResource:Handle) : LONGINT;
OpenResFile	Opens named resource file, makes it current	I	FUNCTION OpenResFile (fileName:Str255) : INTEGER;
OpenRFPerm	Opens resource file with permissions	IV	FUNCTION OpenRFPerm (fileName:Str255; vRefNum:INTEGER; permission:Byte) : INTEGER;
ReleaseResource	Releases memory occupied by resource	I	PROCEDURE ReleaseResource (theResource:Handle);
ResError	Returns error code of previous resource call	I	FUNCTION ResError : INTEGER;
RGetResource	Looks for resource in open files and ROM	V	FUNCTION RGetResource (theType:ResType; theID: INTEGER) : Handle;
RmveResource	Removes resource reference data from resource file	I	PROCEDURE RmveResource (theResource:Handle);
RsrcMapEntry	Gets offset of resource from map	IV	FUNCTION RsrcMapEntry (theResource:Handle) : LONGINT;
RsrcZoneInit	Called automatically by system to init res map	I	PROCEDURE RsrcZoneInit;
SetResAttrs	Changes resource attributes	I	PROCEDURE SetResAttrs (theResource:Handle; attrs:INTEGER);
SetResFileAttrs	Sets resource file attributes	I	PROCEDURE SetResFileAttrs (refNum:INTEGER; attrs:INTEGER);
SetResInfo	Changes ID and name of resource	I	PROCEDURE SetResInfo (theResource:Handle; theID:INTEGER; name:Str255);
SetResLoad	Changes resource load state	I	PROCEDURE SetResload (load:BOOLEAN);
SetResPurge	Sets write state for all resChanged resources	I	PROCEDURE SetResPurge (install:BOOLEAN);
SizeResource	Returns size in bytes of resource	I	FUNCTION SizeResource (theResource:Handle) : LONGINT;
Unique1ID	Returns unique resource ID# for current res file	IV	FUNCTION Unique1ID (theType:ResType) : INTEGER;
UniqueID	Returns unique resource ID number	I	FUNCTION UniqueID (theType:ResType) : INTEGER;
UpdateResFile	Makes resource changes permanent	I	PROCEDURE UpdateResFile (refnum:INTEGER);
UseResFile	Sets current resource file	I	PROCEDURE UseResFile (refNum:INTEGER);
WriteResource	Writes resource data to resource file	I	PROCEDURE WriteResource (theResource:Handle);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-114 to I-127, IV-15 to IV-17, V-30

See Also: 3.010. Resource Manager Result Codes and Resource Attributes

3.010. Resource Manager Result Codes and Resource Attributes

Resource Attributes

Mask Value	*Constant*	*Description*
64	resSysHeap	Set on read into system heap
32	resPurgeable	Set if resource is purgeable
16	resLocked	Set if resource is locked
8	resProtected	Set if resource is protected
4	resPreload	Set if resource is to be preloaded
2	resChanged	Set if resource is to be written to resource file

Resource File Attributes

Mask Value	*Constant*	*Description*
128	mapReadOnly	Set if file is read-only
64	mapCompact	Set if file is to be compacted on update
32	mapChanged	Set if map is to be written on update

Result Codes

Code	*Constant*	*Description*
-192	resNotFound	Resource was not found
-193	resFNotFound	Resource file was not found
-194	addResFailed	AddResource call failed
-196	rmvResFailed	RmveResource call failed
-198	resAttrErr	Attribute was inconsistent with operation requested
-199	mapReadErr	Map was inconsistent with operation requested

Source: Inside Macintosh, pages I-134, IV-21

See Also: 3.009. Resource Manager Routines

3.011 QuickDraw Regions/Pixels/Pictures Routines

Name	*Description*	*Defined **	*Pascal Definition*
AddPt	Adds coordinates of srcPt to dstPt	I	PROCEDURE AddPt (srcPt:Point; VAR dstPt:Point);
BackPixPat	Sets background pattern to multicolor pixel pattern	V	PROCEDURE BackPixPat (ppat:PixPatHandle);
CalcMask	Creates new bitmap as if MacPaint lasso used	IV	PROCEDURE CalcMask (srcPtr,dstPtr:Ptr; srcRow,dstRow,height,words:INTEGER);
ClosePicture	Closes picture definition	I	PROCEDURE ClosePicture;
ClosePoly	Stops defining polygon	I	PROCEDURE ClosePoly;
CloseRgn	Stops collecting region data and saves to dstRgn	I	PROCEDURE CloseRgn (dstRgn:RgnHandle);
CopyBits	Transfers bit image between two bitmaps with clipping	I	PROCEDURE CopyBits (srcBits,dstBits:BitMap; srcRect,dstRect:Rect; mode:INTEGER; maskRgn:RgnHandle);
CopyMask	Creates new bitmap using source's 1 bits	IV	PROCEDURE CopyMask (srcBits,maskBits,dstBits:BitMap; srcRect,maskRect,dstRect:Rect);
CopyPixMap	Duplicates pixMap	V	PROCEDURE CopyPixMap (srcPM,dstPM:PixMapHandle);
CopyPixPat	Duplicates pixPat	V	PROCEDURE CopyPixPat (srcPP,dstPP:PixPatHandle);
CopyRgn	Makes duplicate copy of srcRgn	I	PROCEDURE CopyRgn (srcRgn,dstRgn:RgnHandle);
DiffRgn	Creates new region by subtracting RgnB from RgnA	I	PROCEDURE DiffRgn (srcRgnA,srcRgnB,dstRgn:RgnHandle);
DisposPixMap	Releases storage used by pixMap	V	PROCEDURE DisposPixMap (pm:PixMapHandle);
DisposPixPat	Releases storage used by pixPat	V	PROCEDURE DisposPixPat (ppat:PixPatHandle);
DrawPicture	Draws picture in dstRect	I	PROCEDURE DrawPicture (myPicture:PicHandle; dstRect:Rect);
EmptyRect	Returns TRUE if rectangle is empty	I	FUNCTION EmptyRect (r:Rect) : BOOLEAN;
EmptyRgn	Returns TRUE if region is empty	I	FUNCTION EmptyRgn (rgn:RgnHandle) : BOOLEAN;
EqualPt	Returns TRUE if two points equal	I	FUNCTION EqualPt (pt1,pt2:Point) : BOOLEAN;
EqualRect	Returns TRUE if two rects equal	I	FUNCTION EqualRect (rect1,rect2:Rect) : BOOLEAN;
EqualRgn	Returns TRUE if two regions identical	I	FUNCTION EqualRgn (rgnA, rgnB:RgnHandle) : BOOLEAN;
EraseArc	Paints inside of arc within rectangle with background pattern **	I	PROCEDURE EraseArc (r:Rect; startAngle,arcAngle:INTEGER);
EraseOval	Paints inside of oval within rectangle with background pattern **	I	PROCEDURE EraseOval (r:Rect);
ErasePoly	Paints inside of polygon with background pattern **	I	PROCEDURE ErasePoly (poly:PolyHandle);
EraseRect	Paints inside of rectangle with background pattern **	I	PROCEDURE EraseRect (r:Rect);
EraseRgn	Paints inside of region with background pattern **	I	PROCEDURE EraseRgn (rgn:RgnHandle);
EraseRoundRect	Paints inside of rounded rectangle with background pattern **	I	PROCEDURE EraseRoundRect (r:Rect; ovalWidth,ovalHeight:INTEGER);
FillArc	Fills arc within rectangle with pattern	I	PROCEDURE FillArc (r:Rect; ; startAngle,arcAngle:INTEGER; pat:Pattern);
FillOval	Fills oval within rectangle with pattern	I	PROCEDURE FillOval (r:Rect; pat:Pattern);
FillPoly	Fills polygon with pattern	I	PROCEDURE FillPoly (poly:PolyHandle; pat:Pattern);
FillRect	Fills rectangle with pattern	I	PROCEDURE FillRect (r:Rect; pat:Pattern);
FillRgn	Fills region with pattern	I	PROCEDURE FillRgn (rgn:RgnHandle; pat:Pattern);
FillRoundRect	Fills rounded rectangle with pattern	I	PROCEDURE FillRoundRect (r:Rect; ; ovalWidth,ovalHeight:INTEGER; pat:Pattern);
FrameArc	Draws outline inside arc within rectangle **	I	PROCEDURE FrameArc (r:Rect; startAngle,arcAngle:INTEGER);
FrameOval	Draws outline inside oval within rectangle **	I	PROCEDURE FrameOval (r:Rect);
FramePoly	Draws outline inside polygon **	I	PROCEDURE FramePoly (poly:PolyHandle);
FrameRect	Draws outline inside rectangle **	I	PROCEDURE FrameRect (r:Rect);
FrameRgn	Draws outline inside region **	I	PROCEDURE FrameRgn (rgn:RgnHandle);
FrameRoundRect	Draws outline inside rounded rectangle **	I	PROCEDURE FrameRoundRect (r:Rect; ovalWidth,ovalHeight:INTEGER);
GetPixPat	Returns handle to new pixPat created with ppat	V	FUNCTION GetPixPat (patID:INTEGER) : PixPatHandle;
GlobalToLocal	Converts point from global coordinate system to local	I	PROCEDURE GlobalToLocal (VAR pt:Point);
InsetRect	Changes rectangle size	I	PROCEDURE InsetRect (VAR r:Rect; dh,dv:INTEGER);
InsetRgn	Changes region size	I	PROCEDURE InsetRgn (rgn:RgnHandle; dh,dv:INTEGER);
InvertArc	Inverts pixels enclosed by wedge of rectangle	I	PROCEDURE InvertArc (r:Rect; startAngle,arcAngle:INTEGER);
InvertOval	Inverts pixels of oval enclosed by rectangle	I	PROCEDURE InvertOval (r:Rect);
InvertPoly	Inverts pixels enclosed by polygon	I	PROCEDURE InvertPoly (poly:PolyHandle);
InvertRect	Inverts pixels enclosed by rectangle	I	PROCEDURE InvertRect (r:Rect);
InvertRgn	Inverts pixels enclosed by region	I	PROCEDURE InvertRgn (rgn:RgnHandle);
InvertRoundRect	Inverts pixels enclosed by rounded rectangle	I	PROCEDURE InvertRoundRect (r:Rect; ovalWidth,ovalHeight:INTEGER);
KillPicture	Releases memory occupied by picture	I	PROCEDURE KillPicture (myPicture:PicHandle);
KillPoly	Releases memory occupied by polygon	I	PROCEDURE KillPoly (poly:PolyHandle);
LocalToGlobal	Converts point to global coordinate system	I	PROCEDURE LocalToGlobal (VAR pt:Point);

(continued)

Table 3.011. Continued

Name	*Description*	*Defined **	*Pascal Definition*
MakeRGBPat	Creates pixPat with approximate RGB color	V	PROCEDURE MakeRGBPat (ppat:PixPatHandle; myColor:RGBColor);
NewPixMap	Returns handle to new pixMap structure	V	FUNCTION NewPixMap : PixMapHandle;
NewPixPat	Returns handle to new pixPat structure	V	FUNCTION NewPixPat:PixPatHandle;
NewRgn	Allocates space for new region	I	FUNCTION NewRgn : RgnHandle;
OffsetPoly	Moves polygon on coordinate plane	I	PROCEDURE OffsetPoly (poly:PolyHandle; dh,dv:INTEGER);
OffsetRect	Moves rectangle by dh, dv amounts	I	PROCEDURE OffsetRect (VAR r:Rect; dh,dv:INTEGER);
OffsetRgn	Moves region on coordinate plane	I	PROCEDURE OffsetRgn (rgn:RgnHandle; dh,dv:INTEGER);
OpenPicture	Returns handle to new picture	I	FUNCTION OpenPicture (picFrame:Rect) : PicHandle;
OpenPoly	Returns handle to new polygon, starts saving def	I	FUNCTION OpenPoly : PolyHandle;
OpenRgn	Allocates space and collects data for new region	I	PROCEDURE OpenRgn;
PaintArc	Paints inside of arc within rectangle **	I	PROCEDURE PaintArc (r:Rect; startAngle,arcAngle:INTEGER);
PaintOval	Paints inside of oval within rectangle **	I	PROCEDURE PaintOval (r:Rect);
PaintPoly	Paints inside of polygon **	I	PROCEDURE PaintPoly (poly:PolyHandle);
PaintRect	Paints inside of rectangle **	I	PROCEDURE PaintRect (r:Rect);
PaintRgn	Paints inside of region **	I	PROCEDURE PaintRgn (rgn:RgnHandle);
PaintRoundRect	Paints inside of rounded rectangle **	I	PROCEDURE PaintRoundRect (r:Rect; ovalWidth,ovalHeight:INTEGER);
PenPixPat	Sets pen pattern to multicolor pixel pattern	V	PROCEDURE PenPixPat (ppat:PixPatHandle);
PicComment	Inserts comment into current picture	I	PROCEDURE PicComment (kind,dataSize:INTEGER; dataHandle:Handle);
Pt2Rect	Generates smallest rectangle that encloses two points	I	PROCEDURE Pt2Rect (pt1,pt2:Point; VAR dstRect:Rect);
PtInRect	Returns TRUE if point is enclosed in rectangle	I	FUNCTION PtInRect (pt:Point; r:Rect) : BOOLEAN;
PtInRgn	Returns TRUE if point is in region	I	FUNCTION PtInRgn (pt:Point; rgn:RgnHandle) : BOOLEAN;
PtToAngle	Calculates angle straight up from center of rectangle to point	I	PROCEDURE PtToAngle (r:Rect; pt:Point; VAR angle:INTEGER);
RectInRgn	Returns TRUE if rectangle intersects region	I	FUNCTION RectInRgn (r:Rect; rgn: RgnHandle) : BOOLEAN;
RectRgn	Destroys previous region def, sets to new rectangle	I	PROCEDURE RectRgn (rgn:RgnHandle; r:Rect);
ScrollRect	Scrolls bits of rectangle with grafPort	I	PROCEDURE ScrollRect (r:Rect; dh,dv:INTEGER; updateRgn:RgnHandle);
SectRect	Generates rectangle intersection of two rectangles	I	FUNCTION SectRect (src1,src2:Rect; VAR dstRect:Rect) : BOOLEAN;
SectRgn	Creates new region from intersection of two regions	I	PROCEDURE SectRgn (srcRgnA,srcRgnB,dstRgn:RgnHandle);
SeedFill	Creates new bitmap with MacPaint-like fill	IV	PROCEDURE SeedFill (srcPtr,dstPtr:Ptr; srcRow,dstRow,height,words,seedH,seedV:INTEGER);
SetEmptyRgn	Destroys previous region def, then same as NewRgn	I	PROCEDURE SetEmptyRgn (rgn:RgnHandle);
SetPt	Assigns coordinates to point	I	PROCEDURE SetPt (VAR Pt:Point; h,v:INTEGER);
SetRect	Assigns boundary coordinates for rectangle	I	PROCEDURE SetRect (VAR r:Rect; left,top,right,bottom:INTEGER);
SetRectRgn	Destroys previous region def, sets to new coordinates	I	PROCEDURE SetRectRgn (rgn:RgnHandle; left,top,right,bottom:INTEGER);
SubPt	Subtracts coordinates of srcPt from dstPt	I	PROCEDURE SubPt (srcPt:Point; VAR dstPt:Point);
UnionRect	Generates smallest rectangle that encloses two rectangles	I	PROCEDURE UnionRect (src1,src2:Rect; VAR dstRect:Rect);
UnionRgn	Creates new region from union of two regions	I	PROCEDURE UnionRgn (srcRgnA, srcRgnB,dstRgn:RgnHandle);
XorRgn	Creates new region by XORing union and intersection of two rectangles	I	PROCEDURE XorRgn (srcRgnA,srcRgnB,dstRgn:RgnHandle);

Notes: •* Volume of Inside Macintosh where first definition of routine appeared (I–V).
•** Uses settings of current grafPort (pnMode, pnPat, and so on).

Source: Inside Macintosh, pages I-174 to I-194, IV-23 to IV-25, V-70 to V-75

See Also: 3.024. QuickDraw Bitmap Data Structure
3.026. QuickDraw Picture Data Structure
3.027. QuickDraw Polygon Data Structure
3.028. QuickDraw Pen State Data Structure
3.031. QuickDraw Pen Transfer Modes
3.034. Color QuickDraw Routines
3.039. PixMap Data Structure
3.040. PixPat Data Structure

3.012. QuickDraw Miscellaneous Routines

Name	*Description*	*Defined **	*Pascal Definition*
GetPixel	Returns TRUE if coordinate point is black	I	FUNCTION GetPixel (h,v:INTEGER) : BOOLEAN;
MapPoly	Maps (scaled) polygon to destination rectangle	I	PROCEDURE MapPoly (poly: PolyHandle; srcRect,dstRect:Rect);
MapPt	Maps (scaled) pt to destination rectangle	I	PROCEDURE MapPt (VAR pt:Point; srcRect,dstRect:Rect);
MapRect	Maps (scaled) rectangle to destination rectangle	I	PROCEDURE MapRect (VAR r:Rect;srcRect,dstRect:Rect);
MapRgn	Maps (scaled) region to destination rectangle	I	PROCEDURE MapRgn (rgn:RgnHandle;srcRect,dstRect:Rect);
Random	Returns pseudo random integer	I	FUNCTION Random : INTEGER;
ScalePt	Scales point to new size	I	PROCEDURE ScalePt (VAR pt:Point; srcRect,dstRect:Rect);
StuffHex	Stores bits into a data structure	I	PROCEDURE StuffHex (thingPtr:Ptr; s:Str255);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-194 to I-197

See Also: 3.011. QuickDraw Regions/Pixels/Pictures Routines
3.013. QuickDraw grafPort Routines
3.014. QuickDraw Pen/Line Drawing Routines
3.015. QuickDraw Cursor Routines
3.016. QuickDraw Text Drawing Routines
3.021. Point Data Structure
3.022. Rect Data Structure
3.023. Region Data Structure
3.027. Polygon Data Structure
3.034. Color QuickDraw Routines

3.013. QuickDraw GrafPort Routines

Name	*Description*	*Defined **	*Pascal Definition*
BackPat	Sets background pattern for current grafPort	I	PROCEDURE BackPat (pat:Pattern);
ClipRect	Changes clipping region of current grafPort to rectangle	I	PROCEDURE ClipRect (r:Rect);
CloseCPort	Closes color grafPort; releases assigned memory	V	PROCEDURE CloseCPort (port:CGrafPtr);
ClosePort	Closes grafPort; releases assigned memory	I	PROCEDURE ClosePort (port:GrafPtr);
GetClip	Makes region equivalent to current grafPort clipping region	I	PROCEDURE GetClip (rgn:RgnHandle);
GetPort	Returns pointer to current grafPort	I	PROCEDURE GetPort (port:GrafPtr);
GrafDevice **	Sets device field for current grafPort	I	PROCEDURE GrafDevice (device: INTEGER);
InitCPort	Reinitializes previously opened color grafPort	V	PROCEDURE InitCPort (port:CGrafPtr);
InitGraf	Initializes QuickDraw	I	PROCEDURE InitGraf (globalPtr:Ptr);
InitPort	Reinitializes previously opened grafPort	I	PROCEDURE InitPort (port:GrafPtr);
MovePortTo **	Changes position of current grafPort's portRect	I	PROCEDURE MovePortTo (leftGlobal,topGlobal:INTEGER);
OpenCPort	Opens new color grafPort for use	V	PROCEDURE OpenCPort (port:CGrafPtr);
OpenPort	Opens new grafPort for use	I	PROCEDURE OpenPort (port:grafPtr);
PortSize **	Changes size of current grafPort's portRect	I	PROCEDURE PortSize (width, height:INTEGER);
SetClip	Changes clipping region of current grafPort to region	I	PROCEDURE SetClip (rgn:RgnHandle);
SetOrigin	Changes local coordinate system of grafPort	I	PROCEDURE SetOrigin (h,v:INTEGER);
SetPort	Specifies current grafPort	I	PROCEDURE SetPort (port:GrafPtr);
SetPortBits	Sets portBits field for current grafPort to bitmap	I	PROCEDURE SetPortBits (bm:BitMap);

Notes: •* Volume of Inside Macintosh where first definition of routine appeared (I–V).
•** Routine normally used internally by Toolbox, not by applications.

Source: Inside Macintosh, pages I-162 to I-167, V-67 to V-68

See Also: 3.011. QuickDraw Regions/Pixels/Pictures Routines
3.012. QuickDraw Miscellaneous Routines
3.014. QuickDraw Pen/Line Drawing Routines
3.015. QuickDraw Cursor Routines
3.016. QuickDraw Text Drawing Routines
3.019. QuickDraw GrafPort Data Structure
3.033. QuickDraw InitGraf Default Settings
3.038. ColorGrafPort Data Structure

3.014. QuickDraw Pen/Line Drawing Routines

Name	*Description*	*Defined **	*Pascal Definition*
GetPen	Returns current pen position	I	PROCEDURE GetPen (VAR pt:Point);
GetPenState	Saves current pen state in pnState	I	PROCEDURE GetPenState (VAR pnState:PenState);
HidePen	Decrements pnVis (negative = don't draw)	I	PROCEDURE HidePen;
Line	Moves pen and draws line indicated distance	I	PROCEDURE Line (dh,dv:INTEGER);
LineTo	Moves pen and draws line to location	I	PROCEDURE LineTo (h,v:INTEGER);
Move	Moves pen indicated distance	I	PROCEDURE Move (dh,dv:INTEGER);
MoveTo	Moves pen to location	I	PROCEDURE MoveTo (h,v:INTEGER);
PenMode	Sets pen transfer mode	I	PROCEDURE PenMode (mode:INTEGER);
PenNormal	Resets pen state to defaults	I	PROCEDURE PenNormal;
PenPat	Sets pen pattern	I	PROCEDURE PenPat (pat:Pattern);
PenSize	Sets dimensions of pen	I	PROCEDURE PenSize (width,height:INTEGER);
SetPenState	Sets current pen state from pnState	I	PROCEDURE SetPenState (pnState:PenState);
ShowPen	Increments pnVis (0 = draw)	I	PROCEDURE ShowPen;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-168 to I-171

See Also: 3.011. QuickDraw Regions/Pixels/Pictures Routines
3.012. QuickDraw Miscellaneous Routines
3.013. QuickDraw GrafPort Routines
3.014. QuickDraw Pen/Line Drawing Routines
3.015. QuickDraw Cursor Routines
3.028. QuickDraw Pen State Data Structure
3.031. QuickDraw Pen Transfer Modes

3.015. QuickDraw Cursor Routines

Name	*Description*	*Defined **	*Pascal Definition*
AllocCursor	Reallocates cursor memory	V	PROCEDURE AllocCursor;
DisposCCursor	Disposes of all CCrsr structures	V	PROCEDURE DisposCCursor (cCrsr:CCrsrHandle);
GetCCursor	Creates new CCrsr structure	V	FUNCTION GetCCursor (crsrID:INTEGER) :CCrsrHandle;
HideCursor	Decrements cursor level (negative = invisible)	I	PROCEDURE HideCursor;
InitCursor	Sets cursor to arrow and level to 0 (visible)	I	PROCEDURE InitCursor;
ObscureCursor	Hides cursor until next mouse movement	I	PROCEDURE ObscureCursor;
SetCCursor	Sets application cursor to colored cursor	V	PROCEDURE SetCCursor (cCrsr:CCrsrHandle);
SetCursor	Sets new cursor type	I	PROCEDURE SetCursor (crsr:Cursor);
ShowCursor	Increments cursor level (0 = visible)	I	PROCEDURE ShowCursor;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-167 to I-168, V-75

See Also: 3.011. QuickDraw Regions/Pixels/Pictures Routines
3.012. QuickDraw Miscellaneous Routines
3.013. QuickDraw GrafPort Routines
3.014. QuickDraw Pen/Line Drawing Routines
3.016. QuickDraw Text Drawing Routines
3.018. Cursor Data Structure
3.043. Color QuickDraw Cursor Data Structure

3.016. QuickDraw Text Drawing Routines

Name	*Description*	*Defined **	*Pascal Definition*
CharWidth	Returns width of specified character	I	FUNCTION CharWidth (ch:CHAR) : INTEGER;
DrawChar	Draws character to right of pen location	I	PROCEDURE DrawChar (ch:Char);
DrawString	Draws string to right of pen location with no formatting	I	PROCEDURE DrawString (s:Str255);
DrawText	Draws text block in memory as per DrawString	I	PROCEDURE DrawText (textBuf:Ptr; firstByte,byteCount:INTEGER);
GetFontInfo	Returns state of current grafPort's font	I	PROCEDURE GetFontInfo (VAR info: FontInfo);
MeasureText	Array version of TextWidth	IV	PROCEDURE MeasureText (count:INTEGER; textAddr,charLocs:Ptr);
SpaceExtra	Sets current grafPort's spExtra field	I	PROCEDURE SpaceExtra (extra:Fixed);
StringWidth	Returns width of specified string	I	FUNCTION StringWidth (s:Str255) : INTEGER;
TextFace	Sets current grafPort's character style	I	PROCEDURE TextFace (face:Style);
TextFont	Sets current grafPort's font	I	PROCEDURE TextFont (font:INTEGER);
TextMode	Sets current grafPort's transfer mode	I	PROCEDURE TextMode (mode:INTEGER);
TextSize	Sets current grafPort's font size	I	PROCEDURE TextSize (size:INTEGER);
TextWidth	Returns width of specified text in memory	I	FUNCTION TextWidth (textBuf:Ptr; firstByte,byteCount:INTEGER) : INTEGER;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-171 to I-173, IV-25

See Also: 3.011. QuickDraw Regions/Pixels/Pictures Routines
3.012. QuickDraw Miscellaneous Routines
3.013. QuickDraw GrafPort Routines
3.014. QuickDraw Pen/Line Drawing Routines
3.015. QuickDraw Cursor Routines
3.020. QuickDraw Text Style (StyleItem) Equates
3.030. QuickDraw FontInfo Data Structure
3.031. QuickDraw Pen Transfer Modes

3.017. QuickDraw Constants

Value	*Constant*	*Comments*
0	srcCopy	Source transfer mode
1	srcOr	Source transfer mode
2	srcXor	Source transfer mode
3	srcBic	Source transfer mode
4	notSrcCopy	Source transfer mode
5	notSrcOr	Source transfer mode
6	notSrcXor	Source transfer mode
7	notSrcBic	Source transfer mode
8	patCopy	Pattern transfer mode
9	patOr	Pattern transfer mode
10	patXor	Pattern transfer mode
11	patBic	Pattern transfer mode
12	notPatCopy	Pattern transfer mode
13	notPatOr	Pattern transfer mode
14	notPatXor	Pattern transfer mode
15	notPatBic	Pattern transfer mode
33	blackColor	Standard ForeColors/BackColors
30	whiteColor	Standard ForeColors/BackColors
205	redColor	Standard ForeColors/BackColors
341	greenColor	Standard ForeColors/BackColors
409	blueColor	Standard ForeColors/BackColors
273	cyanColor	Standard ForeColors/BackColors
137	magentaColor	Standard ForeColors/BackColors
69	yellowColor	Standard ForeColors/BackColors
0	boldBit	Text style
1	italicBit	Text style
2	ulineBit	Text style
3	outlineBit	Text style
5	shadowBit	Text style
6	extendBit	Text style
0	picLParen	Standard picComment
1	picRParen	Standard picComment

Source: Inside Macintosh, pages I-152, I-158 to I-159, I-169 to I-170, I-188, I-201

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.031. QuickDraw Pen Transfer Modes

3.018. Cursor Data Structure

Size	*Name*	*Comments*
16 Bits16	Data	Cursor image arranged as 16x16 bit array *
16 Bits16	Mask **	Cursor mask arranged as 16x16 bit array *
Point	HotSpot	Point to align with mouse position

Notes: •* First Bits16 contains first line of bits, second Bits16 contains second line, and so on.
•** Mask and data interact to create visible cursor as follows:

Data	*Mask*	*Resulting Pixel*
0	1	White
1	1	Black
0	0	Same as pixel under cursor
1	0	Inverse of pixel under cursor

Source: Inside Macintosh, page I-146 to I-147

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.031. QuickDraw Pen Transfer Modes

3.019. QuickDraw GrafPort Data Structure

Size	*Name*	*Comments*
INTEGER	device	Device-specific information
BitMap	portBits	Bitmap for grafPort
Rect	portRect	Rectangle for grafPort
RgnHandle	visRgn	Visible region of grafPort
RgnHandle	clipRgn	Clipping region of grafPort
Pattern	bkPat	Background pattern
Pattern	fillpat	Fill pattern
Point	pnLoc	Pen location
Point	pnSize	Pen size
INTEGER	pnMode	Pen's transfer mode
Pattern	pnPat	Pen pattern
INTEGER	pnVis	Pen visibility
INTEGER	txFont	Font number for text
Style	txFace	Style for text
INTEGER	txMode	Transfer mode for text
INTEGER	txSize	Size for text
Fixed	spExtra	Extra space
LONGINT	fgColor	Foreground color
LONGINT	bkColor	Background color
INTEGER	colrBit	Color bit
INTEGER	patStretch	USED INTERNALLY
Handle	picSave	Picture being saved
Handle	rgnSave	Region being saved
Handle	polySave	Polygon being saved
QDProcsPtr	grafProcs	Pointer to low-level drawing routines

Source: Inside Macintosh, pages I-147 to I-155

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.020. QuickDraw Text Style (StyleItem) Equates
3.021. QuickDraw Point Data Structure
3.022. QuickDraw Rect Data Structure
3.023. QuickDraw Region Data Structure
3.024. QuickDraw Bitmap Data Structure
3.031. QuickDraw Pen Transfer Modes

3.020. QuickDraw Text Style (StyleItem) Equates

Value	*Constant*	*Comments*
0	boldBit	Emboldened text
1	italicBit	Italicized text
2	ulineBit	Underlined text
3	outlineBit	Outlined text
5	shadowBit	Drop-shadowed text
6	extendBit	Extended-width text

Notes:
- Equates are bit numbers defined for assembly language programmers.
- Other languages use StyleItem set (bold, italic, underline, outline, shadow, condense, extend).

Source: Inside Macintosh, page I-152

See Also: 3.011 to 3.016, 3.034 QuickDraw routine tables

3.021. QuickDraw Point Data Structure

Size	*Name*	*Comments*
INTEGER	v	Vertical coordinate
INTEGER	h	Horizontal coordinate

Source: Inside Macintosh, page I-202

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

3.022. QuickDraw Rect Data Structure

Size	*Name*	*Comments*
INTEGER	Top	Top coordinate of rectangle
INTEGER	Left	Left coordinate of rectangle
INTEGER	Bottom	Bottom coordinate of rectangle
INTEGER	Right	Right coordinate of rectangle

Alternate Definition

Point	TopLeft	Upper left corner of rectangle
Point	BottomRight	Bottom right corner of rectangle

Source: Inside Macintosh, page I-202

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

3.023. QuickDraw Region Data Structure

Size	*Name*	*Comments*
INTEGER	rgnSize	Size of region (in bytes) *
Rect	rgnBBox	Rectangle enclosing region
varies	data	Additional data if region is not rectangular **

Notes:
- * Maximum size = 32K bytes.
- ** Compacted for efficient access by QuickDraw.

Source: Inside Macintosh, pages I-141 to I-143

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.022. QuickDraw Rect Data Structure

3.024. QuickDraw Bitmap Data Structure

Size	*Name*	*Comments*
Ptr	baseAddr	Pointer to base address of bit image *
INTEGER	rowBytes	Width of one row of image (in pixels) *
Rect	bounds	Rectangle that encloses active area of bitmap

Notes: * Must be an even number to start on word boundary.

Source: Inside Macintosh, pages I-144 to I-145

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

3.025. QuickDraw Color Equates (Old Style)

Value	*Constant*
33	blackColor
30	whiteColor
205	redColor
341	greenColor
409	blueColor
273	cyanColor
137	magentaColor
69	yellowColor

Notes: These colors have been supplemented by RGB color in the Mac II (256K ROM)

Source: Inside Macintosh, page I-158

See Also: 3.035. RGBColor Data Structure
3.036. Color Table Data Structure
3.037. ColorSpec Data Structure
3.149. ColorInfo Data Structure
3.150. Palette Data Structure

3.026. QuickDraw Picture Data Structure

Size	*Name*	*Comments*
INTEGER	picSize	Size of the picture (in bytes) *
Rect	picFrame	Rectangle that surrounds picture
Varies	data	QuickDraw commands and picComments

Notes: • Data consists of picComments and PICT opcodes; see 2.087. to 2.090.
•* Maximum size is 32K bytes.

Source: Inside Macintosh, pages I-158 to I-159

See Also: 2.087. PICT File Format
2.088. PicComments
2.089. PICT Data Types
2.090. PICT Opcodes
3.011. to 3.016., 3.034. QuickDraw Routines
3.031. QuickDraw Pen Transfer Modes

3.027. QuickDraw Polygon Data Structure

Size	*Name*	*Comments*
INTEGER	polySize	Size of the polygon (in bytes) *
Rect	polyBBox	Rectangle that surrounds entire polygon
Varies	polyPoints	Array of points

Notes: * Maximum size is 32K bytes.

Source: Inside Macintosh, pages I-159 to I-160

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.021. QuickDraw Point Data Structure
3.022. QuickDraw Rect Data Structure

3.028. QuickDraw Pen State Data Structure

Size	*Name*	*Comments*
Point	pnLoc	Location of pen
Point	pnSize	Size of pen (in pixels)
INTEGER	pnMode	Transfer mode used for pen
Pattern	pnPat	Pattern used for pen

Source: Inside Macintosh, page I-169

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.031. QuickDraw Pen Transfer Modes

3.029. QuickDraw QDProcs Data Structure

Size	*Name*	*Values/Comments*
Ptr	textProc	Pointer to text drawing procedure
Ptr	lineProc	Pointer to line drawing procedure
Ptr	rectProc	Pointer to rectangle drawing procedure
Ptr	rRectProc	Pointer to rounded rectangle drawing procedure
Ptr	ovalProc	Pointer to oval drawing procedure
Ptr	arcProc	Pointer to arc drawing procedure
Ptr	polyProc	Pointer to polygon drawing procedure
Ptr	rgnProc	Pointer to region drawing procedure
Ptr	bitsProc	Pointer to bit transfer procedure
Ptr	commentProc	Pointer to picture comment procedure
Ptr	txMeasProc	Pointer to text width measurement procedure
Ptr	getPicProc	Pointer to picture retrieval procedure
Ptr	putPicProc	Pointer to picture saving procedure

Source: Inside Macintosh, page I-197

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

3.030. QuickDraw FontInfo Data Structure

Size	*Name*	*Values/Comments*
INTEGER	ascent	Font ascent
INTEGER	descent	Font descent
INTEGER	widMax	Maximum character width for font
INTEGER	leading	Font leading

Source: Inside Macintosh, page I-173

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

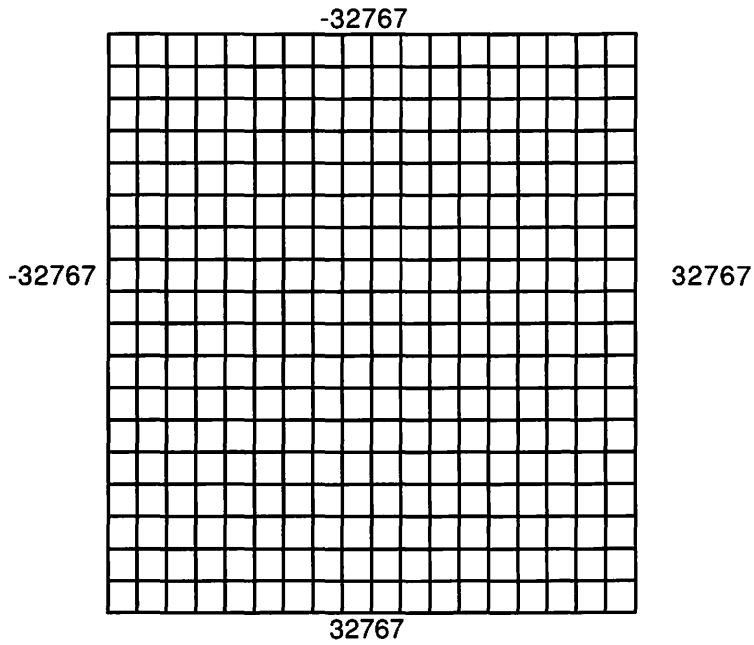
3.031. QuickDraw Pen Transfer Modes

Transfer Mode			
Pattern	*Source*	*Black Pixels in Pattern/Source*	*White Pixels in Pattern/Source*
patCopy	srcCopy	Forced black	Forced white
patOr	srcOr	Forced black	Left alone
patXor	srcXor	Inverted	Left alone
patBic	srcBic	Forced white	Left alone
notPatCopy	notSrcCopy	Forced white	Forced black
notPatOr	notSrcOr	Left alone	Forced black
notPatXor	notSrcXor	Left alone	Inverted
notPatBic	notSrcBic	Left alone	Forced white

Source: Inside Macintosh, pages I-156 to I-157

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

3.032. QuickDraw Coordinate Plane



Notes: Grid lines are assumed to be infinitely thin.

Source: Inside Macintosh, pages I-138 to I-139

3.033. QuickDraw InitGraf Default Settings

Type	*Name*	*Setting*
GrafPtr	thePort	NIL
Pattern	white	All white
Pattern	black	All black
Pattern	gray	50% gray
Pattern	ltGray	25% gray
Pattern	dkGray	75% gray
Cursor	arrow	Standard arrow cursor
BitMap	screenBits	Entire screen
LONGINT	randSeed	1

Source: Inside Macintosh, pages I-162 to I-163

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables

3.034. Color QuickDraw Routines

Name	*Description*	*Defined **	*Pascal Definition*
AllocCursor	Reallocates cursor memory	V	PROCEDURE AllocCursor;
BackPixPat	Same as backPat, but uses multicolor pixel patterns	V	PROCEDURE BackPixPat (ppat: PixPatHandle);
CalcCMask	Generates mask in pixMap (srcBits)	V	PROCEDURE CalcCMask (srcBits,dstBits:BitMap; srcRect,dstRect:Rect; seedRGB:RGBColor; matchProc:ProcPtr; matchData:LONGINT);
CharExtra	Sets CharExtra field in cGrafPort	V	PROCEDURE CharExtra (extra:Fixed);
CloseCPort	Releases memory allocated to cGrafPort	V	PROCEDRUE CloseCport (port:CGrafPtr);
CopyBits	Transfers image from one bitMap/pixMap to another	I **	PROCEDURE CopyBits (srcBits,dstBits:BitMap; srcRect,dstRect:Rect; mode:INTEGER; maskRgn:RgnHandle);
CopyMask	Transfers image using 1 bits as mask	IV **	PROCEDURE CopyMask (srcBits,maskBits,dstBits:BitMap; srcRect,maskRect,dstRect:Rect);
CopyPixMap	Duplicates pixMap structure	V	PROCEDURE CopyPixMap (srcPM,dstPM:PixMapHandle);
CopyPixPat	Copies source pixPat to destination pixPat	V	PROCEDURE CopyPixPat (srcPP,destPP:PixPatHandle);
DisposCCursor	Releases all storage used by CCrsr	V	PROCEDURE DisposCCursor (cCrsr:CCrsrHandle);
DisposCIcon	Releases all storage used by CIcon	V	PROCEDURE DisposCIcon (theIcon:CIconHandle);
DisposCTable	Releases all storage used by CTable	V	PROCEDURE DisposCTable (cTable:CTabHandle);
DisposPixMap	Releases all storage used by pixMap	V	PROCEDURE DisposPixMap (pm:PixMapHandle);
DisposPixPat	Releases all storage used by pixPat	V	PROCEDURE DisposPixPat (ppat:PixPatHandle);
FillCArc	Same as fillArc, but uses multicolor pixel pattern	V	PROCEDURE FillCArc (r:rect; startAngle, arcAngle:INTEGER; ppat:PixPatHandle);
FillCOval	Same as fillOval, but uses multicolor pixel pattern	V	PROCEDURE FillCOval (r:Rect; ppat:PixPatHandle);
FillCPoly	Same as fillPoly, but uses multicolor pixel pattern	V	PROCEDURE FillCPoly (poly: PolyHandle; ppat:PixPatHandle);
FillCRect	Same as fillRect, but uses multicolor pixel pattern	V	PROCEDURE FillCRect (r:Rect; ppat:PixPatHandle);
FillCRgn	Same as fillRgn, but uses multicolor pixel pattern	V	PROCEDURE FillCRgn (rgn: RgnHandle; ppat: PixPatHandle);
FillCRoundRect	Same as fillRoundRect, but uses multicolor pixel pattern	V	PROCEDURE FillCRoundRect (r:rect; ovWd,ovHt:INTEGER; ppat:PixPatHandle);
GetBackColor	Returns current port's RGB background color	V	PROCEDURE GetBackColor (VAR color:RGBColor);
GetCCursor	Returns handle to CCrsr struct, made using 'crsr' resource	V	FUNCTION GetCCursor (crsrID:INTEGER) : CCrsrHandle;
GetCIcon	Returns handle to CIcon struct made using 'cicn' resource	V	FUNCTION GetCIcon (id:INTEGER) : CIconHandle;
GetCPixel	Returns RGB value of pixel at position in port	V	PROCEDURE GetCPixel (h,v:INTEGER; VAR cPix:RGBColor);
GetCTable	Returns handle to new CTable, made using 'clut' resource	V	FUNCTION GetCTable (ctID:INTEGER) : CTabHandle;
GetForeColor	Returns current port's RGB foreground color	V	PROCEDURE GetForeColor (VAR color:RGBColor);
GetPixPat	Returns handle to new pixPat struct using 'ppat' resource	V	FUNCTION GetPixPat (patID:INTEGER) : PixPatHandle;
HiliteColor	Sets highlight color	V	PROCEDURE HiliteColor (color:RGBColor);
InitCPort	Initializes all fields in cGrafPort to default values	V	PROCEDURE InitCPort (port:CGrafPtr);
MakeRGBPat	Generates pixPat with approximate RGB color	V	PROCEDURE MakeRGBPat (ppat:PixPatHandle; myColor:RGBColor);
NewPixPat	Returns handle to new pixPat structure	V	FUNCTION NewPixPat:PixPatHandle;
NewPixMap	Returns handle to new pixMap data structure	V	FUNCTION NewPixMap : PixMapHandle;
OpColor	Sets optimum color for AddPin, SubPin, blend modes	V	PROCEDURE OpColor (color:RGBColor);
OpenCPort	Opens cGrafPort instead of GrafPort	V	PROCEDURE OpenCPort (port:CGrafPtr);
PenPixPat	Same as penPat, but uses multicolor pixel pattern	V	PROCEDURE PenPixPat (ppat:PixPatHandle);
PlotCIcon	Draws CIcon in rectangle	V	PROCEDURE PlotCIcon (theRect:Rect; theIcon:CIconHandle);
RGBBackColor	Sets background to best available matching color for device	V	PROCEDURE RGBBackColor (color:RGBColor);
RGBForeColor	Sets foreground to best available matching color for device	V	PROCEDURE RGBForeColor (color:RGBColor);
SeedCFill	Generates mask for use with CopyBits/CopyMask	V	PROCEDURE SeedCFill (srcBits,dstBits:BitMap; srcRect,dstRect:Rect; seedH,seedV:INTEGER; matchProc:ProcPtr;matchData:LONGINT);
SetCCursor	Sets cursor to multicolored cursor	V	PROCEDURE SetCCursor (cCrsr:CCrsrHandle);
SetCPixel	Changes RGB value of pixel at position in port	V	PROCEDURE SetCPixel (h,v:INTEGER; cPix:RGBColor);
SetPortPix	cGrafPort version of SetPortBits	V	PROCEDURE SetPortPix (pm:PixMapHandle);
SetStdCProcs	Sets all fields of CQDProcs record to low-level routines	V	PROCEDURE SetStdCProcs (VAR cProcs:CQDProcs);

Notes: •* Volume of Inside Macintosh where first definition of routine appeared (I–V).
•** Routine has been changed to handle more data types.

Source: Inside Macintosh, pages V-67 to V-78

See Also: 3.011. to 3.016. QuickDraw routine tables

3.035. RGBColor Data Structure

Size	*Name*	*Comments*
INTEGER	red	Red component of color
INTEGER	green	Green component of color
INTEGER	blue	Blue component of color

Source: Inside Macintosh, page V-48

See Also: 3.034. Color QuickDraw Routines

3.036. Color Table Data Structure

Size	*Name*	*Comments*
LONGINT	ctSeed	Unique ID from table
INTEGER	ctFlags	Flags for SpecArray *
INTEGER	ctSize	Number of entries in table minus 1
cSpecArray	ctTable	Array of color specifiers (see 3.037.)

Notes: * Differentiate between device color table and image color table.

Source: Inside Macintosh, pages V-48 to V-49

See Also: 3.034. Color QuickDraw Routines
3.037. ColorSpec Data Structure

3.037. ColorSpec Data Structure

Size	*Name*	*Values/Comments*
INTEGER	value	Pixel value
RGBColor	rgb	RGB color associated with pixel value *

Notes: * See 3.035. RGBColor Data Structure for format.

Source: Inside Macintosh, page V-49

See Also: 3.034. Color QuickDraw Routines
3.035. RGBColor Data Structure

3.038. ColorGrafPort Data Structure

Size	*Name*	*Values/Comments*
INTEGER	device	Device ID
PixMapHandle	portPixMap	Port's pixel map, describes cGrafPort's pixels (see 3.039.)
INTEGER	portVersion	Two HO bits always set for cGrafPort (not set = regular GrafPort; see 3.019.)
Handle	grafVars	Additional fields (see 3.045.)
INTEGER	chExtra	Fixed point at which to widen every proportionally spaced character
INTEGER	pnLocHFrac	Fractional horizontal pen position (initial value is 1/2)
Rect	portRect	Pixel pattern structure (see 3.040.)
RgnHandle	visRgn	Visible region
RgnHandle	clipRgn	Clipping region
PixPatHandle	bkPixPat	Background pixel pattern
RGBColor	rgbFgColor	Requested foreground color (see 3.035.)
RGBColor	rgbBkColor	Requested background color (see 3.035.)
Point	pnLoc	Pen location
Point	pnSize	Pen size
INTEGER	pnMode	Pen transfer mode
PixPatHandle	pnPixPat	Pen pattern
PixPatHandle	fillPixPat	Fill pattern
INTEGER	pnVis	Pen visible flag
INTEGER	txFont	Font number
Style	txFace	Font style
INTEGER	txMode	Font transfer mode
INTEGER	txSize	Font size
Fixed	spExtra	Extra space
LONGINT	fgColor	Actual foreground color supplied by color manager (see 3.035.)
LONGINT	bkColor	Actual background color supplied by color manager (see 3.035.)
INTEGER	colrBit	Color plane being drawn *
INTEGER	patStretch	USED INTERNALLY
Handle	picSave	Picture being saved
Handle	rgnSave	Region being saved
Handle	polySave	Polygon being saved
CQDProcsPtr	grafProcs	Low-level drawing routines **

Notes:
- * Points to color routines, not same routines as in GrafPort.
- ** RESERVED; not for use by applications.

Source: Inside Macintosh, pages V-49 to V-52

See Also:
- 3.019. QuickDraw GrafPort Data Structure
- 3.034. Color QuickDraw Routines
- 3.035. RGBColor Data Structure
- 3.039. PixMap Data Structure
- 3.040. PixPat Data Structure
- 3.045. GrafVars Data Structure

3.039. PixMap Data Structure

Size	*Name*	*Values/Comments*
Ptr	baseAddr	Pointer to pixMap data *
INTEGER	rowBytes	Offset to next row **
Rect	bounds	Bounding rectangle (in pixels for pixMap)
INTEGER	pmVersion	Color QuickDraw version number (initial release = 0)
INTEGER	packType	Packing format ***
LONGINT	packSize	Size of data in its packed state; must be 0 if packType = 0
Fixed	hRes	Horizontal resolution of pixMap data, in pixels per inch
Fixed	vRes	Vertical resolution of pixMap data, in pixels per inch
INTEGER	pixelType	Storage format for pixel image ****
INTEGER	pixelSize	Number of bits per pixel (always a power of 2)
INTEGER	cmpCount	Number of color components per physical pixel (1 for chunky)
INTEGER	cmpSize	Number of color components per logical pixel
LONGINT	planeBytes	Offset from one plane to next, in bytes (0 in chunky images)
CTabHandle	pmTable	Table of colors used in pixMap (see 3.036.)
LONGINT	pmReserved	RESERVED, must be 0

Notes: •* Multiple of four for optimal performance.
•** Must be even value; three HO bits used as flags:
Bit 15 = 1 means data structure is pixMap, 0 means bitMap.
Bits 14,13 not used; must be 0.
•*** 0 = no packing used to compress image data.
•**** 0 = chunky, 1 = chunky/planar, 2 = planar.

Source: Inside Macintosh, pages V-52 to V-54

See Also: 3.034. Color QuickDraw Routines
3.036. Color Table Data Structure

3.040. PixPat Data Structure

Size	*Name*	*Values/Comments*
INTEGER	patType	Pattern type: 0 = old style, 1 = full color pixel pattern, 2 = RGB pattern
PixMapHandle	patMap	Pixel map
Handle	patData	Pattern's pixel image
Handle	patXData	USED INTERNALLY, expanded pixel image used by Color QuickDraw
INTEGER	patXValid	Validity of data and color (-1 = invalidate)
Handle	patXMap	RESERVED, used by Color QuickDraw
Pattern	pat1Data	8x8 old-style pattern for use with grafPort *

Notes: * Set to 50% gray by NewPixPat.

Source: Inside Macintosh, pages V-55 to V-57

See Also: 3.034. Color QuickDraw Routines
3.039. PixMap Data Structure

3.041. Color QuickDraw Transfer Modes

Mode	*Foreground*	*Background*
Copy	Foreground color applied to black pixels	Background color applied to white pixels
NotCopy	Foreground color applied to white pixels	Background color applied to black pixels
Or	Foreground color applied to black pixels	*
NotOr	Foreground color applied to white pixels	**
Bic	*	Background color applied to black pixels
NotBic	**	Background color applied to white pixels
Xor	Complements bits of black pixels ***	Complements bits of black pixels ***
NotXor	Inverts bits of white pixels ***	Inverts bits of white pixels ***

Notes:
- "Black pixels" are equivalent to 1 bits; "white pixels" are equivalent to 0 bits.
- In all cases, colored source is copied to destination.
- * White part of source is not copied to destination.
- ** Black part of source is not copied to destination.
- *** When used on colored destination, color of destination isn't defined.

Source: Inside Macintosh, pages V-57 to V-59

See Also: 3.034. Color QuickDraw Routines
3.040. PixPat Data Structure

3.042. Color QuickDraw Arithmetic Drawing Modes

Mode	*Value*	*Destination Equals*	*Reverts To* *
addOver	34	Sum of the source and destination RGB values **	Xor
addPin	33	Sum of the destination RGB values, pinned to a maximum ***	Bic
subOver	38	Difference of source and destination RGB values ****	Xor
subPin	35	Difference of the destination RGB values, pinned to a maximum ***	Or
adMax	37	Maximum of source and destination pixels	Bic
adMin	39	Minimum of source and destination pixels	Or
blend	32	Weighted average of source and destination pixel colors †	Copy

Notes:
- * On 1-bit bitmaps, arithmetic mode is approximated by old transfer modes.
- ** Wraps around at 65,536 to 0.
- *** For grafPorts, maximum is white, minimum is black; for cGrafPorts, assigned by OpColor.
- **** Wraps around at 0 to 65,536.
- † Destination = source * weight/65,536 + destination * (1 - weight/65,536).

Source: Inside Macintosh, pages V-59 to V-61

See Also: 3.034. Color QuickDraw Routines

3.043. Color QuickDraw Cursor Data Structure

Size	*Name*	*Comments*
INTEGER	crsrType	Type of cursor; $8000 = old cursor, $8001 = new cursor
PixMapHandle	crsrMap	Pixel map defining cursor's characteristics
Handle	crsrData	Cursor's pixel data
Handle	crsrXData	Private, expanded pixel image used by Color QuickDraw
INTEGER	crsrXValid	Depth of expanded cursor image, set to 0 only, when cursor data changed
Handle	crsrXHandle	RESERVED
Bits16	crsr1Data	16x16 bit image for cursor on 1 and 2-bit per pixel screens
Bits16	crsrMask	16x16 mask image for cursor on 1 and 2-bit per pixel screens
Point	crsrHotSpot	Cursor's hot spot coordinates
LONGINT	crsrXTable	RESERVED
LONGINT	crsrID	ctSeed for cursor

Source: Inside Macintosh, pages V-63 to V-64

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.039. PixMap Data Structure

3.044. Color QuickDraw Icon Data Structure

Size	*Name*	*Comments*
PixMap	iconPMap	Pixel map describing icon (inline structure same as 3.039.)
BitMap	iconMask	Bitmap of the mask for the icon
BitMap	iconBMap	Bitmap of the icon
Handle	iconData	Icon's pixel image
INTEGER array	iconMaskData	Icon's mask data followed by icon's bitmap data *

Notes: * Used only when icon is stored as a resource.

Source: Inside Macintosh, pages V-64 to V-65

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.039. PixMap Data Structure

3.045. GrafVars Data Structure

Size	*Name*	*Comments*
RGBColor	rgbOpColor	Color to use for addPin, subPin, blend (see 3.042.)
RGBColor	rgbHiliteColor	Color for highlighting
Handle	pmFgColor	Foreground color palette
INTEGER	pmFgIndex	Index value for foreground color
Handle	pmBkColor	Background color palette
INTEGER	pmBkIndex	Index value for background color
INTEGER	pmFlags	Palette manager flags

Source: Inside Macintosh, page V-67

See Also: 3.011. to 3.016., 3.034. QuickDraw routine tables
3.042. Color QuickDraw Arithmetic Drawing Modes

3.046. RGB Equivalents of Old QuickDraw Colors

		RGB Color Equivalent		
Value	*Old Color Name*	*Red*	*Green*	*Blue*
0	blackColor	$0000	$0000	$0000
1	yellowColor	$FC00	$F37D	$052F
2	magentaColor	$F2D7	$0856	$84EC
3	redColor	$DD6B	$08C2	$06A2
4	cyanColor	$0241	$AB54	$EAFF
5	greenColor	$0000	$8000	$11B0
6	blueColor	$0000	$0000	$D400
7	whiteColor	$FFFF	$FFFF	$FFFF

Source: Inside Macintosh, page V-69

See Also: 3.034. Color QuickDraw Routines

3.047. QuickDraw Additive Color Table

Primary Colors Used			
Red	*Green*	*Blue*	*Result*
			Black
√			Red
	√		Green
		√	Blue
√	√		Yellow
√		√	Magenta
	√	√	Cyan
√	√	√	White

Source: Technical Introduction to the Macintosh Family, page 100

See Also: 3.035. RGBColor Data Structure

3.048. Color Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
Color2Index	Returns best available appoximation of absolute color	V	FUNCTION Color2Index (rgb:RGBColor) : LONGINT;
Index2Color	Returns RGB color corresponding to color table index	V	PROCEDURE Index2Color (index:LONGINT; VAR rgb:RGBColor);
InvertColor	Complements absolute color	V	PROCEDURE InvertColor (VAR theColor:RGBColor);
RealColor	Returns TRUE if absolute color exists in device's color table	V	FUNCTION RealColor (color:RGBColor) : BOOLEAN;
GetSubTable	Maps RGB values into nearest match in target table	V	PROCEDURE GetSubTable (myColors:CTabHandle; iTabRes:INTEGER; targetTbl:CTabHandle);
MakeITable	Generates inverse table for color table	V	PROCEDURE MakeITable (colorTab:CTabHandle; inverseTab:ITabHandle; res:INTEGER);
GetCTSeed	Returns unique seed value for ctSeed field of color table	V	FUNCTION GetCTSeed : LONGINT;
ProtectEntry	Protects/unprotects entry in color table	V	PROCEDURE ProtectEntry (index:INTEGER; protect:BOOLEAN);
ReserveEntry	Reserves/dereserves entry in color table	V	PROCEDURE ReserveEntry (index:INTEGER; reserve:BOOLEAN);
SetEntries	Sets group of color table entries for current gDevice	V	PROCEDURE SetEntries (start,count:INTEGER; aTable:CSpecArray);
SaveEntries	Saves selected entries from srcTable to resultTable	V	PROCEDURE SaveEntries (srcTable:CTabHandle; ResultTable:CTabHandle; VAR selection:ReqListRec);
RestoreEntries	Sets selected entries from srcTable to dstTable	V	PROCEDURE RestoreEntries (srcTable:CTabHandle;DstTable:CTabHandle; VAR selection:RecListRec);
QDError	Returns last QuickDraw or Color Manager error	V	FUNCTION QDError : INTEGER;
SearchProc	Default search (color match) procedure	V	FUNCTION SearchProc (rgb:RGBColor; VAR position:LONGINT) : BOOLEAN;
CompProc	Default complement search (color match) procedure	V	PROCEDURE CompProc (VAR rgb:RGBColor);
AddSearch	Adds custom procedure to search list (added at head of list)	V	PROCEDURE AddSearch (searchProc:ProcPtr);
AddComp	Adds custom procedure to complement list (added at head of list)	V	PROCEDURE AddComp (compProc:ProcPtr);
DelSearch	Removes custom procedure from search list	V	PROCEDURE DelSearch (searchProc:ProcPtr);
DelComp	Removes custom procedure from complement list	V	PROCEDURE DelComp (compProc:ProcPtr);
SetClientID	Sets gdID field in current device record	V	PROCEDURE SetClientID (id:INTEGER);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-141 to V-147

See Also: 3.008. Interface Toolbox Managers
3.034. Color QuickDraw Routines
3.035. RGBColor Data Structure

3.049. ReqListRec Data Structure

Size	*Name*	*Comments*
INTEGER	reqLSize	Size of request list in items minus 1
INTEGER array	reqLData	Request list data

Source: Inside Macintosh, page V-144

See Also: 3.048. Color Manager Routines

3.050. SProcRec Data Structure

Size	*Name*	*Comments*
SProcHandle	nxtSrch	Handle to the next sProcRec
ProcPtr	srchProc	Pointer to a search procedure

Source: Inside Macintosh, page V-146

See Also: 3.048. Color Manager Routines

3.051. CProcRec Data Structure

Size	*Name*	*Comments*
CProcHandle	nxtComp	Handle to the next cProcRec
ProcPtr	compProc	Pointer to a complement procedure

Source: Inside Macintosh, page V-146

See Also: 3.048. Color Manager Routines

3.052. Font Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
FMSwapFont	Returns pointer to font output record	I	FUNCTION FMSwapFont (inRec:FMInput) : FMOutPtr;
FontMetrics	Same as GetFontInfo (QD), except returns fixed-point values	IV	PROCEDURE FontMetrics (VAR theMetrics:FMetricRec);
GetFNum	Returns Num of font with font name	I	PROCEDURE GetFNum (fontName:Str255; VAR theNum:INTEGER);
GetFontName	Returns name of font with fontNum ID	I	PROCEDURE GetFontName (fontNum:INTEGER; VAR theName:Str255);
InitFonts	Initializes font manager	I	PROCEDURE InitFonts;
RealFont	Returns TRUE if font is available in the asked-for size	I	FUNCTION RealFont (fontNum:INTEGER; size:INTEGER) : BOOLEAN;
SetFontLock	Makes font unpurgeable if TRUE, releases memory used by font if FALSE	I	PROCEDURE SetFontLock (lockFlag:BOOLEAN);
SetFractEnable **	Enables/disables fractional character widths	IV	PROCEDURE SetFractEnable (fractEnable:BOOLEAN);
SetFScaleDisable	Enables/disables font scaling	IV	PROCEDURE SetFScaleDisable (fontScaleDisable:BOOLEAN);

Notes: •* Volume of Inside Macintosh where first definition of routine appeared (I–V).
•** Not in 128K ROMs; in 256K ROMs only.

Source: Inside Macintosh, pages I-222 to I-223, IV-32

See Also: 3.008. Interface Toolbox Managers
3.016. QuickDraw Text Drawing Routines

3.053. Font Numbers

Value	*Constant*	*Comments*
0	SystemFont	System font (normally Chicago)
1	ApplFont	Application font (normally Geneva)
2	NewYork	
3	Geneva	
4	Monaco	
5	Venice	
6	London	
7	Athens	
8	SanFran	
9	Toronto	
11	Cairo	
12	LosAngeles	
20	Times	
21	Helvetica	
22	Courier	
23	Symbol	
24	Taliesin	

Source: Inside Macintosh, page I-219

See Also: 2.072. Font ID Numbers

3.054. FontRec Data Structure

Size	*Name*	*Comments*
INTEGER	fontType	Font type *
INTEGER	firstChar	First character ASCII value in font
INTEGER	lastChar	Last character ASCII value in font
INTEGER	widMax	Maximum character width, in pixels
INTEGER	kernMax	Maximum kerning allowed, in pixels (negative)
INTEGER	nDescent	Font descent, in pixels (negative)
INTEGER	fRectWidth	Font width, in pixels
INTEGER	fRectHeight	Font height, in pixels
INTEGER	owTLoc	Offset from here to offset/width table for font (in this structure)
INTEGER	ascent	Ascent of font, in pixels
INTEGER	descent	Descent of font, in pixels
INTEGER	leading	Leading of font, in pixels
INTEGER	rowWords	Width of row in bit image, divided by 2
INTEGER array	bitImage	Bit image array
INTEGER array	locTable	Location table array
INTEGER array	owTable	Offset/width table array

Notes: * Font types are defined as follows:

Value	*Bit Numbers*																
	15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	
	1																RESERVED
		X															1 = font not to be expanded (Mac II only)
			X														1 = fixed-width, 0 = proportional width font
				1													RESERVED
					0	0											RESERVED
							X										1 = font contains colors (Mac II only)
								X									1 = synthetic font (Mac II only)
									X								1 = font has fctb resource (Mac II only)
										0	0	0					RESERVED
													X	X			Font depth (Mac II only)
															X		1 = has character-width table
																X	1 = has image-height table
$ACB0	1	0	1	0	1	1	0	0	1	0	1	1	0	0	0	0	Font width data (64K ROM only)

Source: Inside Macintosh, pages I-230 to I-232, IV-35, V-184 to V-185

See Also: 3.052. Font Manager Routines

3.055. FMInput Data Structure

Size	*Name*	*Comments*
INTEGER	family	Font number
INTEGER	size	Font size
Style	face	Font style
BOOLEAN	needBits	TRUE indicates characters will be drawn
INTEGER	device	Device-specific information
Point	numer	Numerator of scaling factor
Point	denom	Denominator of scaling factor

Source: Inside Macintosh, pages I-224 to I-225

See Also: 3.021. QuickDraw Point Data Structure
3.052. Font Manager Routines

3.056. FMOutput Data Structure

Size	*Name*	*Comments*
INTEGER	errNum	NOT USED
Handle	fontHandle	Handle to the font record
Byte	bold	Bold factor *
Byte	italic	Italic factor *
Byte	ulOffset	Underline offset *
Byte	ulShadow	Underline shadow *
Byte	ulThick	Underline thickness *
Byte	shadow	Shadow factor *
SignedByte	extra	Width of style
Byte	ascent	Ascent
Byte	descent	Descent
Byte	widMax	Maximum character width
SignedByte	leading	Leading
Byte	UNUSED	
Point	numer	Numerators of scaling factor
Point	denom	Denominators of scaling factor

Notes: * Factors taken from 3.057. Font Characterization Table.

Source: Inside Macintosh, page I-227

See Also: 3.021. QuickDraw Point Data Structure
3.052. Font Manager Routines
3.057. Font Characterization Table

3.057. Font Characterization Table

Size	*Description*	*Value for Display Screen*
Word	Pixels per vertical inch on device	80
Word	Pixels per horizontal inch on device	80
3 bytes *	Bold characteristics	0,1,1
3 bytes *	Italic characteristics	1,8,0
3 bytes *	NOT USED	0,0,0
3 bytes *	Outline characteristics	5,1,1
3 bytes *	Shadow characteristics	5,2,2
3 bytes *	Condensed characteristics	0,0,-1
3 bytes *	Extended characteristics	0,0,1
3 bytes **	Underline characteristics	1,1,1

Notes: •* First byte indicates byte number after bold field in FMOutput being affected. Second byte contains number to be stored in field. Third byte contains amount to increment extra field of FMOutput.
•** First byte: underline offset; second byte: underline shadow; third byte: underline thickness.

Source: Inside Macintosh, page I-226

See Also: 3.021. QuickDraw Point Data Structure
3.052. Font Manager Routines
3.056. FMOutput Data Structure

3.058. Font Metrics Data Structure

Size	*Name*	*Description*
Fixed	ascent	Ascent of font
Fixed	descent	Descent of font
Fixed	leading	Leading of font
Fixed	widMax	Maximum character width
Handle	wTabHandle	Handle to global width table

Source: Inside Macintosh, page IV-32

See Also: 3.030. QuickDraw FontInfo Data Structure
3.052. Font Manager Routines

3.059. FamRec Data Structure

Size	*Name*	*Comments*
INTEGER	ffFlags *	Family flags
INTEGER	ffFamID	Font family ID number
INTEGER	ffFirstChar	Value of ASCII code for first character in font
INTEGER	ffLastChar	Value of ASCII code for last character in font
INTEGER	ffAscent	Maximum ascent for 1-point font
INTEGER	ffDescent	Maximum descent for 1-point font
INTEGER	ffLeading	Maximum leading for 1-point font
INTEGER	ffWidMax	Maximum width for 1-point font
LONGINT	ffWTabOff	Offset to width table (ffWidthTab)
LONGINT	ffKernOff	Offset to kerning table (ffKernTab)
LONGINT	ffStylOff	Offset to style-mapping table (ffStyTab)
INTEGER array	ffProperty	Style property information
INTEGER array	ffIntl	RESERVED
INTEGER	ffVersion	Version number
Varies	ffAssoc	Font association table
Varies	ffWidthTab	Font width table
Varies	ffStyTab	Font style-mapping table
Varies	ffKernTab	Font kerning table

Notes: * Family flags defined as follows:

Bit Numbers																
15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	
X																1 = fixed-width, 0 = proportional-width font
	X															1 = fractional width table not used, 0 = fractional width table used
		X														1 = use integer extra for stylistic variations
			X													1 = ignore FractEnable, 0 = use FractEnable
				0	0	0	0	0	0	0	0	0	0			RESERVED
														X		1 = has character-width table
															X	1 = has image-height table

Source: Inside Macintosh, pages IV-36 to IV-37

See Also: 3.052. Font Manager Routines

3.060. WidthTable Data Structure

Size	*Name*	*Comments*
Fixed array	tabData	Character width table (256 entries)
Handle	tabFont	Font record used to build width table
LONGINT	sExtra	Space extra used for table
LONGINT	style	Extra due to style
INTEGER	fID	Font family ID
INTEGER	fSize	Font size request
INTEGER	face	Style request
INTEGER	device	Device requested
Point	inNumer	Numerators of scaling factors
Point	inDenom	Denominators of scaling factors
INTEGER	aFID	Actual font family ID for table
Handle	fHand	Family record used to build width table
BOOLEAN	usedFam	TRUE if fixed-point family widths used
Byte	aFace	Actual face produced
INTEGER	vOutput	Vertical factor for expanding characters
INTEGER	hOutput	Horizontal factor for expanding characters
INTEGER	vFactor	NOT USED
INTEGER	hFactor	Horizontal factor for increasing character widths
INTEGER	aSize	Actual size of font used
INTEGER	tabSize	Total size of table

Source: Inside Macintosh, page IV-41

See Also: 3.052. Font Manager Routines
3.054. FontRec Data Structure

3.061. Script Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
Char2Pixel	Returns screen position of carets and selection points	V	FUNCTION Char2Pixel (textBuf:Ptr; textLength,slop,offset:INTEGER; direction:SignedByte) : INTEGER;
CharByte	Returns character type of byte at offset (see 3.063.)	V	FUNCTION CharByte (textBuf:Ptr; textOffset:INTEGER) : INTEGER;
CharType	Returns information about byte at offset (see 3.063.)	V	FUNCTION CharType (textBuf:Ptr; textOffset:INTEGER) : INTEGER;
DrawJust	Draws text similar to QuickDraw DrawText, adds slop pixels at end	V	PROCEDURE DrawJust (textPtr:Ptr; textLength,slop:INTEGER);
FindWord	Returns two offsets in table which specify boundaries of word	V	PROCEDURE FindWord (textPtr:Ptr; textLength,offset:INTEGER; leftSide:BOOLEAN; breaks:BreakTable; VAR offsets:OffsetTable);
Font2Script	Translates font ID to script code	V	FUNCTION Font2Script (fontNumber:INTEGER) : INTEGER;
FontScript	Returns script code for font script	V	FUNCTION FontScript:INTEGER;
GetAppFont	Returns ID number of current application font	V	FUNCTION GetAppFont:INTEGER;
GetDefFontSize	Returns size of current default font	V	FUNCTION GetDefFontSize:INTEGER;
GetEnvirons	Returns values of global Script Manager variables	V	FUNCTION GetEnvirons (verb:INTEGER) : LONGINT;
GetMBarHeight	Returns height of menu bar, in pixels **	V	FUNCTION GetMBarHeight:INTEGER;
GetScript	Returns values of local script variables and routine vectors	V	FUNCTION GetScript (script,verb:INTEGER) : LONGINT;
GetSysFont	Returns ID number of current system font	V	FUNCTION GetSysFont:INTEGER;
GetSysJust	Returns direction in which lines are written ***	V	FUNCTION GetSysJust:INTEGER;
HiliteText	Finds characters between two offsets that should be highlighted	V	PROCEDURE HiliteText (textPtr:Ptr; textLength,firstOffset,secondOffset:INTEGER; VAR offsets:OffsetTable);
IntlScript	Returns script code for International Utilities script	V	FUNCTION IntlScript:INTEGER;
KeyScript	Changes keyboard script	V	PROCEDURE KeyScript (scriptCode:INTEGER);
MeasureJust	Fills array with textWidths of first textLength characters of text at textPtr	V	PROCEDURE MeasureJust (textPtr:Ptr; textLength,slop:INTEGER; charLocs:Ptr);
Pixel2Char	Returns offset in buffer corresponding to a given pixel width	V	FUNCTION Pixel2Char (textBuf:Ptr; textLen,slop,pixelWidth:INTEGER; VAR leftSide:BOOLEAN) : INTEGER;
SetEnvirons	Changes global Script Manager variables	V	FUNCTION SetEnvirons (verb:INTEGER; param:LONGINT) : OSErr;
SetScript	Changes local script variables and routine vectors	V	FUNCTION SetScript (script,verb:INTEGER; param:LONGINT) : OSErr;
SetSysJust	Changes direction in which lines are written ***	V	PROCEDURE SetSysJust (newJust:INTEGER);
Transliterate	Converts text to closest possible approximation in script	V	FUNCTION Transliterate (srcHandle, dstHandle:Handle; target:INTEGER; srcMask:LONGINT) : INTEGER;

Notes:
- * Volume of Inside Macintosh where first definition of routine appeared (I–V).
- ** Cannot be used with 64K ROM.
- *** 0 = left justified (default), -1 = right justified.

Source: Inside Macintosh, pages V-306 to V-316

See Also:
- 3.008. Interface Toolbox Managers
- 3.063. CharByte and CharType Values
- 3.064. Trasliterate Target Values
- 3.065. GetScript Verb Parameter Values
- 3.066. GetEnvirons Verb Parameter Values

3.062. Script ID Numbers

Value	*Constant*	*Comments*
0	smRoman	Normal ASCII alphabet, Roman style
1	smKanji	Japanese
2	smChinese	Chinese
3	smKorean	Korean
4	smArabic	Arabic
5	smHebrew	Hebrew
6	smGreek	Greek
7	smRussian	Cyrillic
8	smReserved1	RESERVED
9	smDevanagari	Devanagari
10	smGurmukhi	Gurmukhi
11	smGujarati	Gujarati
12	smOriya	Oriya
13	smBengali	Bengali
14	smTamil	Tamil
15	smTelugu	Telugu
16	smKannada	Kannada
17	smMalayalam	Malayalam
18	smSinhalese	Sinhalese
19	smBurmese	Burmese
20	smKhmer	Cambodian
21	smThai	Thai
22	smLaotian	Laotian
23	smGeorgian	Georgian
24	smArmenian	Armenian
25	smMaldivian	Maldivian
26	smTibetan	Tibetan
27	smMongolian	Mongolian
28	smAmharic	Ethiopian
29	smSlavic	Non-Cyrillic Slavic
30	smVietnamese	Vietnamese
31	smSindhi	Sindhi
32	smUninterp	Uninterpreted symbols (MacPaint palette tools)

Source: Inside Macintosh, pages V-297 to V-298

See Also: 3.061. Script Manager Routines

3.063. CharByte and CharType Values

CharByte

Value	*Description*
-1	First byte of multibyte character ($FF)
0	Single-byte character
1	Last byte of multibyte character
2	Middle byte of multibyte character

CharType

Bit Number	*Contents*	*Allowable Bit Values*
15	Character size	1 = 2 byte, 0 = 1 byte
14	Character case	1 = uppercase, 0 = lowercase
13	Direction	1 = right to left, 0 = left to right
12	RESERVED	
8 to 11	Character class (subset of char. type)	0000 = punctuation normal 0001 = punctuation number 0010 = punctuation symbol 0011 = punctuation blank
4 to 7	RESERVED	
0 to 3	Character type	0000 = character punctuation 0001 = character ASCII 0111 = character european

Source: Inside Macintosh, pages V-306 to V-307

See Also: 3.061. Script Manager Routines

3.064. Transliterate Target Values

Value	*Constant*	*Description*
0	smTransAscii	Target is roman script
1	smTransNative	Target is non-roman script
2	smTransCase	Switch case for any target
16384	smTransLower	Target becomes lowercase
32768	smTransUpper	Target becomes uppercase
1	smMaskAscii	Convert only roman script
2	smMaskNative	Convert only non-roman script
-1	smMaskAll	Convert all text

Source: Inside Macintosh, page V-311

See Also: 3.061. Script Manager Routines

3.065. GetScript Verb Parameter Values

Value	*Constant*	*Description*
0	smScriptVersion	Software version
2	smScriptMunged	Script entry changed count
4	smScriptEnabled	Script enabled flag
6	smScriptRight	Right-to-left flag
8	smScriptJust	Justification flag
10	smScriptRedraw	Word redraw flag
12	smScriptSysFont	Preferred system font
14	smScriptAppFont	Preferred application font
16	smScriptNumber	Script 'itl0' ID *
18	smScriptDate	Script 'itl1' ID *
20	smScriptSort	Script 'itl2' ID *
22	smScriptRsvd1	RESERVED
24	smScriptRsvd2	RESERVED
26	smScriptRsvd3	RESERVED
28	smScriptRsvd4	RESERVED
30	smScriptRsvd5	RESERVED
32	smScriptKeys	Script 'KEYC' ID *
34	smScriptIcon	Script 'SICN' ID *
36	smScriptPrint	Script printer action routine
38	smScriptTrap	Trap entry pointer
40	smScriptCreator	Script file creator
42	smScriptFile	Script file name
44	smScriptName	Script name

Notes: * Obtained from dictionary.

Source: Inside Macintosh, pages V-312 to V-313

See Also: 3.061. Script Manager Routines

3.066. GetEnvirons Verb Parameter Values

Value	*Constant*	*Description*
0	smVersion	Environment version
2	smMunged	Globals changed count
4	smEnabled	Environment enabled flag
6	smBiDirect	Set if scripts of different directions installed
8	smFontForce	Force font flag
10	smIntlForce	Force international utilities flag
12	smForced	Current script forced to system script
14	smForced	Current script defaulted to roman script
16	smPrint	Printer action routine
18	smSysScript	System script
20	smAppScript	Application script
22	smKeyScript	Keyboard script
24	smSysRef	System folder reference number
26	smKeyCache	Keyboard table cache pointer
28	smKeySwap	Swapping table pointer

Source: Inside Macintosh, page V-313

See Also: 3.061. Script Manager Routines

3.067. Toolbox Event Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
Button	Returns TRUE if button is down, FALSE otherwise	I	FUNCTION Button : BOOLEAN;
EventAvail	Returns TRUE if event of specified type is available **	I	FUNCTION EventAvail (eventMask:INTEGER; VAR theEvent:EventRecord) : BOOLEAN;
GetCaretTime	Returns caret blink time, in ticks	I	FUNCTION GetCaretTime : LONGINT;
GetDblTime	Returns suggested double-click time, in ticks	I	FUNCTION GetDblTime : LONGINT;
GetKeys	Returns keyboard state in keyMap	I	PROCEDURE GetKeys (VAR theKeys:KeyMap);
GetMouse	Returns current mouse location	I	PROCEDURE GetMouse (VAR mouseLoc:Point);
GetNextEvent	Returns next available event of specfied type	I	FUNCTION GetNextEvent (eventMask:INTEGER; VAR theEvent:EventRecord) : BOOLEAN;
KeyTrans	Converts key codes to ASCII values using 'KCHR' resource	V	FUNCTION KeyTrans (transData:Ptr; keycode:INTEGER; VAR state:LONGINT) : LONGINT;
StillDown	Returns TRUE if button is down and no more mouse events in queue	I	FUNCTION StillDown : BOOLEAN;
TickCount	Returns number of ticks since system started	I	FUNCTION TickCount : LONGINT;
WaitMouseUp	Same as StillDown, except removes preceding mouse-up event before returning FALSE	I	FUNCTION WaitMouseUp : BOOLEAN;

Notes: •* Volume of Inside Macintosh where first definition of routine appeared (I–V).
•** Event remains in queue.

Source: Inside Macintosh, pages I-257 to I-260, V-194 to V-195

See Also: 3.008. Toolbox Interface Managers
3.068. Toolbox Event Priorities
3.069. Toolbox EventRecord Data Structure
3.070. Toolbox Event Codes
3.071. Toolbox Event Masks
3.072. Toolbox Event Modifier Flags

3.068. Toolbox Event Priorities

Priority	*Event(s)*	*Comments*
Highest	Activate *	Window inactive before window becomes active
	Mouse-down, mouse-up, key-down, key-up, disk-inserted, network, device driver, application-defined	Priority determined by FIFO order in queue
	Auto-key	
	Update	Done in front-to-back order of windows
Lowest	Null	

Notes: * Detected in special way; not actually placed in event queue.

Source: Inside Macintosh, page I-245

See Also: 3.067. Toolbox Event Manager Routines

3.069. Toolbox EventRecord Data Structure

Size	*Name*	*Comments*
INTEGER	what	See 3.070.
LONGINT	message	*
LONGINT	when	Number of ticks since startup
Point	where	Mouse location
INTEGER	modifiers	See 3.072.

Notes: * Message field varies depending upon what type of event occurred:

Event	*Message field contains*
Keyboard	Character code, key code in LO word, ADB address in HO word
Activate, update	Pointer to window
Disk-inserted	Drive number in LO word, File Manager result code in HO word
Mouse-down, mouse-up, null	UNDEFINED
Network	Handle to parameter block
Device driver	Varies depending upon driver
Application-defined	Varies depending upon application

Source: Inside Macintosh, pages I-249, V-193 to V-194

See Also: 3.067. Toolbox Event Manager Routines
3.070. Toolbox Event Codes
3.072. Toolbox Event Modifier Flags
3.186. File Manager Result Codes

3.070. Toolbox Event Codes

Value	*Event Name*
0	nullEvent
1	mouseDown
2	mouseUp
3	keyDown
4	keyUp
5	autoKey
6	updateEvt
7	diskEvt
8	activateEvt
9	NOT USED
10	networkEvt
11	driverEvt
12	app1Evt
13	app2Evt
14	app3Evt
15	app4Evt

Source: Inside Macintosh, page I-249

See Also: 3.067. Toolbox Event Manager Routines

3.071. Toolbox Event Masks

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	*Description*
X	X	X	X													Application defined
				X												Device driver
					X											Network
						X										RESERVED
							X									Activate
								X								Disk-inserted
									X							Update
										X						Auto-key
											X					Key-up
												X				Key-down
													X			Mouse-up
														X		Mouse-down
															X	NOT USED

Source: Inside Macintosh, pages I-253 to I-254

See Also: 3.067. Toolbox Event Manager Routines

3.072. Toolbox Event Modifier Flags

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	*Description*	*Values*
X	X	X														NOT USED	
			X													Control Key	1 = down, 0 = up
				X												Option Key	1 = down, 0 = up
					X											Caps Lock Key	1 = down, 0 = up
						X										Shift Key	1 = down, 0 = up
							X									Command Key	1 = down, 0 = up
								X								Mouse Button	1 = up, 0 = down
									X	X	X	X	X	X		NOT USED	
															X	Window Activation	1 = activated, 0 = deactivated

Source: Inside Macintosh, pages I-252 to I-253, V-194

See Also: 3.067. Toolbox Event Manager Routines
3.069. Toolbox EventRecord Data Structure

3.073. Window Manager Routines

Name	*Description*	*Defined* *	*Pascal Definition*
BeginUpdate	Updates visRgn of window's grafPort	I	PROCEDURE BeginUpdate (theWindow:WindowPtr);
BringToFront	Brings window to front and redraws windows as necessary	I	PROCEDURE BringToFront (theWindow:WindowPtr);
CalcVis **	Calculates visRgn of given window	I	PROCEDURE CalcVis (window:WindowPeek);
CalcVisBehind **	Calculates visRgns of startWindow and all windows behind it	I	PROCEDURE CalcVisBehind (startWindow:WindowPeek; clobberedRgn:RgnHandle);
CheckUpDate **	Checks from front to back window list and updates as needed	I	FUNCTION CheckUpdate (VAR theEvent:EventRecord) : BOOLEAN;
ClipAbove **	Sets clipRgn of Window Manager port	I	PROCEDURE ClipAbove (window:WindowPeek);
CloseWindow	Removes window from display and window list	I	PROCEDURE CloseWindow (theWindow:WindowPtr);
DisposeWindow	Calls CloseWindow and releases memory used by window	I	PROCEDURE DisposeWindow (theWindow:WindowPtr);
DragGrayRgn	Call when mouse down in region to let WM handle dragging of region	I	FUNCTION DragGrayRgn (theRgn:RgnHandle; startPt:Point; limitRect,slopRect:Rect; axis:INTEGER; actionProc:ProcPtr) : LONGINT;
DragWindow	Handles dragging of window while mouse button down	I	PROCEDURE DragWindow (theWindow:WindowPtr; startPt:Point; boundsRect:Rect);
DrawGrowIcon	Draws grow box area in window	I	PROCEDURE DrawGrowIcon (theWindow:WindowPtr);
DrawNew **	If update = TRUE, updates the area (see source for exact info.)	I	PROCEDURE DrawNew (window:WindowPeek; update:BOOLEAN);
EndUpdate	Restores normal visRgn to window's grafPort after update	I	PROCEDURE EndUpdate (theWindow:WindowPtr);
FindWindow	Returns which part of window mouse event occurred in	I	FUNCTION FindWindow (thePt:Point; VAR whichWindow:WindowPtr) : INTEGER;
FrontWindow	Returns pointer to frontmost window in list	I	FUNCTION FrontWindow : WindowPtr;
GetAuxWin	Returns handle to window's auxiliary window record	V	FUNCTION GetAuxWin (theWindow:WindowPtr; VAR colors:CTabHandle) : BOOLEAN;
GetCWMgrPort	Returns address of WMgrCPort	V	PROCEDURE GetCWMgrPort (VAR wport:CGrafPtr);
GetGrayRgn	Returns handle to grayRgn	V	FUNCTION GetGrayRgn : regionHandle;
GetNewCWindow	Same as GetNewWindow, but creates color window	V	FUNCTION GetNewCWindow (windowID:INTEGER; wStorage:Ptr; behind:CWindowPtr) : CWindowPtr;
GetNewWindow	Creates new window according to window template	I	FUNCTION GetNewWindow (windowID:INTEGER; wStorage:Ptr; behind:WindowPtr) : WindowPtr;
GetWindowPic	Returns handle to picture in window record	I	FUNCTION GetWindowPic (theWindow:WindowPtr) : PicHandle;
GetWMgrPort	Returns pointer to Window Manager port	I	PROCEDURE GetWMgrPort (VAR wPort:GrafPtr);
GetWRefCon	Returns window's reference value	I	FUNCTION GetWRefCon (theWindow:WindowPtr) : LONGINT;
GetWTitle	Returns window's title	I	PROCEDURE GetWTitle (theWindow:WindowPtr; VAR title:Str255);
GetWVariant	Returns variant code for whichWindow	V	FUNCTION GetWVariant (whichWindow:WindowPtr) : INTEGER;
GrowWindow	Handles grow window sizing, returns new size in LONGINT	I	FUNCTION GrowWindow (theWindow:WindowPtr; startPt:Point; sizeRect:Rect) : LONGINT;
HideWindow	Makes window invisible	I	PROCEDURE HideWindow (theWindow:WindowPtr);
HiliteWindow	Highlights/unhighlights window based upon fHilite value	I	PROCEDURE HiliteWindow (theWindow:WindowPtr; fHilite:BOOLEAN);
InitWindows	Initializes Window Manager	I	PROCEDURE InitWindows;
InvalRect	Tells Window Manager that Rect area has changed, needs update	I	PROCEDURE InvalRect (badRect:Rect);
InvalRgn	Tells Window Manager that Rgn has changed, needs update	I	PROCEDURE InvalRgn (badRgn:RgnHandle);
MoveWindow	Moves window on screen	I	PROCEDURE MoveWindow (theWindow:WindowPtr; hGlobal,vGlobal:INTEGER; front:BOOLEAN);
NewCWindow	Same as NewWindow, but creates color window	V	FUNCTION NewCWindow (wStorage:Ptr; boundsRect:Rect; title:Str255; visible:BOOLEAN; procID: INTEGER; behind:WindowPtr; goAwayFlag:BOOLEAN; refCon:LONGINT) : CWindowPtr;
NewWindow	Creates new window according to parameters	I	FUNCTION NewWindow (wStorage:Ptr; boundsRect:Rect; title:Str255; visible:BOOLEAN; procID:INTEGER; behind:WindowPtr; goAwayFlag:BOOLEAN; refCon:LONGINT) : WindowPtr;
PaintBehind **	Calls PaintOne for startWindow and all windows behind it	I	PROCEDURE PaintBehind (startWindow:WindowPeek; clobberedRgn:RgnHandle);
PaintOne **	Paints given window	I	PROCEDURE PaintOne (window:WindowPeek; clobberedRgn:RgnHandle);
PinRect	Returns thePt if in theRect, otherwise returns nearest point in Rect	I	FUNCTION PinRect (theRect:Rect; thePt:Point) : LONGINT;
SaveOld **	Saves window's current structure region and content region	I	PROCEDURE SaveOld (window:WindowPeek);
SelectWindow	Activates window	I	PROCEDURE SelectWindow (theWindow:WindowPtr);
SendBehind	Sends window behind behindWindow, redraws windows	I	PROCEDURE SendBehind (theWindow,behindWindow:WindowPtr);
SetDeskCPat ***	Sets desktop pattern to pattern	V	PROCEDURE SetDeskCPat (deskPixPat:PixPatHandle);
SetWinColor	Sets window's color table	V	PROCEDURE SetWinColor (theWindow:WindowPtr; newColorTable:CTabHandle);
SetWindowPic	Stores picture handle in window record	I	PROCEDURE SetWindowPic (theWindow:WindowPtr; pic:PicHandle);
SetWRefCon	Sets window's reference value	I	PROCEDURE SetWRefCon (theWindow:WindowPtr; data:LONGINT);
SetWTitle	Sets window title, redraws frame	I	PROCEDURE SetWTitle (theWindow:WindowPtr; title:Str255);

(continued)

Table 3.073. Continued

Name	*Description*	*Defined **	*Pascal Definition*
ShowHide	Toggles visibility of window based upon showFlag	I	PROCEDURE ShowHide (theWindow:WindowPtr; showFlag:BOOLEAN);
ShowWindow	Makes window visible	I	PROCEDURE ShowWindow (theWindow:WindowPtr);
SizeWindow	Resizes window	I	PROCEDURE SizeWindow (theWindow:WindowPtr; w,h:INTEGER; fUpdate:BOOLEAN);
TrackBox	Handles zoom box mouse-down event	IV	FUNCTION TrackBox (theWindow:WindowPtr; thePt:Point; partCode:INTEGER) : BOOLEAN;
TrackGoAway	Handles goAway box highlighting, returns goAway status	I	FUNCTION TrackGoAway (theWindow:WindowPtr; thePt:Point) : BOOLEAN;
ValidRect	Tells Window Manager that Rect has been updated	I	PROCEDURE ValidRect (goodRect:Rect);
ValidRgn	Tells Window Manager that Rgn has been updated	I	PROCEDURE ValidRgn (goodRgn:RgnHandle);
window def **	Defines your own window function	I	FUNCTION MyWindow (varCode:INTEGER; theWindow:WindowPtr; message:INTEGER; param:LONGINT) : LONGINT;
ZoomWindow	Called after TrackBox returns TRUE to zoom window	IV	PROCEDURE ZoomWindow (theWindow:WindowPtr; partCode:INTEGER; front:BOOLEAN);

Notes:
- * Volume of Inside Macintosh where first definition of routine appeared (I–V).
- ** Low-level routine; not normally used.
- *** Not used by applications.

Source: Inside Macintosh, pages I-281 to I-299, IV-50, V-206 to V-210

See Also:
3.008. Interface Toolbox Managers
3.074. Window Types
3.075. WindowRecord Data Structure
3.076. CWindowRecord Data Structure
3.077. AuxWinRec Data Structure
3.078. WinCTab Data Structure
3.079. Mouse Locations in Windows
3.080. Window Part Identifiers for Color Tables
3.081. Window Definition Messages
3.082. Window Hit Messages

3.074. Window Types

Value	*Constant*	*Type*	*Used for*
0	documentProc	Standard document window	Application documents
1	dBoxProc	Alert box or modal dialog	Alerts, modals dialogs
2	plainDBox	Plain dialog box	Dialogs
3	altDBoxProc	Plain dialog box with drop shadow	Dialogs
4	noGrowDocProc	Document window without size box	Dialogs
16	rDocProc	Rounded corner window	Desk accessories

Source: Inside Macintosh, pages I-273 to I-274

See Also: 3.073. Window Manager Routines
3.075. WindowRecord Data Structure

3.075. WindowRecord Data Structure

Size	*Name*	*Comments*
GrafPort	port	See 3.019.
INTEGER	windowKind	See 3.074.
BOOLEAN	visible	TRUE if visible
BOOLEAN	hilited	TRUE if highlighted
BOOLEAN	goAwayFlag	TRUE if go-away region included
BOOLEAN	zoomFlag	TRUE if window zoomed
RgnHandle	strucRgn	Structure region
RgnHandle	contRgn	Content region
RgnHandle	updateRgn	Update region
Handle	windowDefProc	Window definition function
Handle	dataHandle	Data used by windowDefProc
StringHandle	titleHandle	Window's title
INTEGER	titleWidth	Width of window title, in pixels
ControlHandle	controlList	Window's control list
WindowPeek	nextWindow	Next window in window list
PicHandle	windowPic	Picture for drawing window
LONGINT	refCon	Window's reference value

Notes: Page IV-52 of Inside Macintosh lists a windowSize entry at the end of the WindowRecord Data Structure, but no size of name is given.

Source: Inside Macintosh, pages I-276 to I-277, IV-52

See Also: 3.019. QuickDraw GrafPort Data Structure
3.073. Window Manager Routines
3.074. Window Types

3.076. CWindowRecord Data Structure

Size	*Name*	*Comments*
CGrafPort	port	See 3.038.
INTEGER	windowKind	See 3.074.
BOOLEAN	visible	TRUE if visible
BOOLEAN	hilited	TRUE if highlighted
BOOLEAN	goAwayFlag	TRUE if go-away region included
BOOLEAN	flagZoom	RESERVED
RgnHandle	strucRgn	Structure region
RgnHandle	contRgn	Content region
RgnHandle	updateRgn	Update region
Handle	windowDefProc	Window definition function
Handle	dataHandle	Data used by windowDefProc
StringHandle	titleHandle	Window's title
INTEGER	titleWidth	Width of window title, in pixels
ControlHandle	controlList	Window's control list
CWindowPeek	nextWindow	Next window in window list
PicHandle	windowPic	Picture for drawing window
LONGINT	refCon	Window's reference value

Source: Inside Macintosh, pages V-199 to V-200

See Also: 3.038. ColorGrafPort Data Structure
3.073. Window Manager Routines
3.074. Window Types
3.075. WindowRecord Data Structure

3.077. AuxWinRec Data Structure

Size	*Name*	*Comments*
AuxWinHandle	awNext	Handle to next record in list
WindowPtr	awOwner	Pointer to owning window
CTabHandle	awCTable	Handle to window's color table
CTabHandle	dialogCItem	RESERVED for Dialog Manager
LONGINT	awFlags	RESERVED
CTabHandle	awReserved	RESERVED
LONGINT	awRefCon	For application use

Source: Inside Macintosh, pages V-200 to V-201

See Also: 3.036. Color Table Data Structure
3.073. Window Manager Routines

3.078. WinCTab Data Structure

Size	*Name*	*Comments*
LONGINT	wCSeed	Unique table ID
INTEGER	wCReserved	NOT USED for windows
INTEGER	ctSize	Number of entries in table minus 1
cSpecArray	ctTable	Array of ColorSpec records

Source: Inside Macintosh, pages V-202 to V-203

See Also: 3.037. ColorSpec Data Structure
3.073. Window Manager Routines

3.079. Mouse Locations in Windows

Value	*Constant*	*Comments*
0	inDesk	Desktop (i.e., none of the other areas)
1	inMenuBar	In menu bar
2	inSysWindow	In system window
3	inContent	In content region (but not in grow region, if active)
4	inDrag	In drag region
5	inGrow	In grow region (of active window only)
6	inGoAway	In go-away region (of active window only)
7	inZoomIn	In zoom box (window zoomed out)
8	inZoomOut	In zoom box (window zoomed in)

Source: Inside Macintosh, pages I-287, IV-51

See Also: 3.073. Window Manager Routines

3.080. Window Part Identifiers for Color Tables

Value	*Constant*	*Comments*
0	wContentColor	Color of content region background
1	wFrameColor	Color of window frame
2	wTextColor	Color of text in content region
3	wHiliteColor	Color of highlighted areas in window
4	wTitleBarColor	Color of title bar

Source: Inside Macintosh, page V-211

See Also: 3.073. Window Manager Routines

3.081. Window Definition Messages

Value	*Constant*	*Comments*
0	wDraw	Draw window frame
1	wHit	Identify where mouse button was pressed
2	wCalcRgns	Calculate strucRgn/contRgn
3	wNew	Perform additional window initialization
4	wDispose	Perform additional disposal tasks
5	wGrow	Draw window's grow image
6	wDrawGIcon	Draw size box in content region

Source: Inside Macintosh, page I-299

See Also: 3.073. Window Manager Routines

3.082. Window Hit Messages

Value	*Constant*	*Comments*
0	wNoHit	Hit, but not in any of the following
1	wInContent	In content region (but not grow area, if active)
2	wInDrag	In drag region
3	wInGrow	In grow region of active window only
4	wInGoAway	In go-away region of active window only
5	wInZoomIn	In zoom box with window zoomed out
6	wInZoomOut	In zoom box with window zoomed in

Source: Inside Macintosh, pages I-300 to I-301, IV-51

See Also: 3.073. Window Manager Routines

3.083. Control Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
DisposeControl	Removes control from screen, releases memory	I	PROCEDURE DisposeControl (theControl:ControlHandle);
DragControl	Pulls dotted outline of control around screen	I	PROCEDURE DragControl (theControl:ControlHandle; startPt:Point; limitRect,slopRect:Rect; axis:INTEGER);
Draw1Control	Draws specified control, if visible in window	IV	PROCEDURE Draw1Control (theControl:ControlHandle);
DrawControls	Draws all visible controls in window	I	PROCEDURE DrawControls (theWindow:WindowPtr);
FindControl	Returns which control, if any, button was pressed in	I	FUNCTION FindControl (thePoint:Point; theWindow:WindowPtr; VAR whichControl:ControlHandle) : INTEGER;
GetAuxCtl	Returns handle to control's color table	V	FUNCTION GetAuxCtl (theControl:ControlHandle; VAR acHndl:AuxCtlHndl) : BOOLEAN;
GetCRefCon	Returns reference value for control	I	FUNCTION GetCRefCon (theControl:ControlHandle) : LONGINT;
GetCTitle	Returns control's title	I	PROCEDURE GetCTitle (theControl:ControlHandle; VAR title:Str255);
GetCtlAction	Returns pointer to control's default action procedure	I	FUNCTION GetCtlAction (theControl:ControlHandle) : ProcPtr;
GetCtlMax	Returns maximum value of control's current setting	I	FUNCTION GetCtlMax (theControl:ControlHandle) : INTEGER;
GetCtlMin	Returns minimum value of control's current setting	I	FUNCTION GetCtlMin (theControl:ControlHandle) : INTEGER;
GetCtlValue	Returns control's current setting	I	FUNCTION GetCtlValue (theControl:ControlHandle) : INTEGER;
GetCVariant	Returns variant control value for control	V	FUNCTION GetCVariant (theControl:ControlHandle) : INTEGER;
GetNewControl	Creates control from template in resource file	I	FUNCTION GetNewControl (controlID:INTEGER; theWindow:WindowPtr) : ControlHandle;
HideControl	Makes control invisible	I	PROCEDURE HideControl (theControl:ControlHandle);
HiliteControl	Changes highlighting of control	I	PROCEDURE HiliteControl (theControl:ControlHandle; hiliteState:INTEGER);
KillControls	Removes all controls associated with window	I	PROCEDURE KillControls (theWindow:WindowPtr);
MoveControl	Moves control to new location in window	I	PROCEDURE MoveControl (theControl:ControlHandle; h,v:INTEGER);
NewControl	Creates and returns handle to new control	I	FUNCTION NewControl (theWindow:WindowPtr; boundsRect:Rect; title:Str255; visible:BOOLEAN; value:INTEGER; min,max:INTEGER; procID:INTEGER; refCon:LONGINT) : ControlHandle
SetCRefCon	Sets control's reference value to the given data	I	PROCEDURE SetCRefCon (theControl:ControlHandle; data:LONGINT);
SetCTitle	Sets control's title	I	PROCEDURE SetCTitle (theControl:ControlHandle; title:Str255);
SetCtlAction	Sets control's default action procedure	I	PROCEDURE SetCtlAction (theControl:ControlHandle; actionProc:ProcPtr);
SetCtlColor	Modifies control's color table	V	PROCEDURE SetCtlColor (theControl:ControlHandle; newColorTable:CCTabHandle);
SetCtlMax	Sets maximum value for control's current setting	I	PROCEDURE SetCtlMax (theControl:ControlHandle; maxValue:INTEGER);
SetCtlMin	Sets minimum value for control's current setting	I	PROCEDURE SetCtlMin (theControl:ControlHandle; minValue:INTEGER);
SetCtlValue	Sets current setting to value and redraws control	I	PROCEDURE SetCtlValue (theControl:ControlHandle; theValue:INTEGER);
ShowControl	Makes control visible	I	PROCEDURE ShowControl (theControl:ControlHandle);
SizeControl	Changes controls enclosing rectangle size	I	PROCEDURE SizeControl (theControl:ControlHandle; w,h:INTEGER);
TestControl	Returns active part code, or 0 if active outside control	I	FUNCTION TestControl (theControl:ControlHandle; thePoint:Point) : INTEGER;
TrackControl	Tracks mouse use within control	I	FUNCTION TrackControl (theControl:ControlHandle; startPt:Point; actionProc:ProcPtr) : INTEGER;
UpdtControl	Faster version of DrawControl	IV	PROCEDURE UpdtControl (theWindow:WindowPtr; updateRgn:RgnHandle);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-319 to I-328, IV-53, V-222

See Also:
3.008. Interface Toolbox Manager Routines
3.084. Control Types
3.085. Control Part Codes
3.086. ControlRecord Data Structure
3.087. AuxCtlRec Data Structure
3.088. CtlCTab Data Structure
3.089. Control Part Identifiers for Color Tables
3.090. Control Axis Parameter Values
3.091. Control Definition Messages

3.084. Control Types

Value	*Constant*	*Used for*
0	pushButProc	Buttons
1	checkBoxProc	Check boxes
2	radioButProc	Radio buttons
16	scrollBarProc	Scroll bars

Source: Inside Macintosh, page I-315

See Also: 3.083. Control Manager Routines
3.086. ControlRecord Data Structure

3.085. Control Part Codes

Value	*Constant*	*Comments*
10	inButton	In button
11	inCheckBox	In check box or radio button
20	inUpButton	In up arrow of scroll bar
21	inDownButton	In down arrow of scroll bar
22	inPageUp	In page up region of scroll bar
23	inPageDown	In page down region of scroll bar
129	inThumb	In thumb of scroll bar

Source: Inside Macintosh, pages I-315 to I-316

See Also: 3.083. Control Manager Routines

3.086. ControlRecord Data Structure

Size	*Name*	*Comments*
ControlHandle	nextControl	Next control in list
WindowPtr	contrlOwner	Control's owning window
Rect	contrlRect	Rectangle that encloses control
Byte	contrlVis	255 if control is visible
Byte	contrlHilite	Highlight state
INTEGER	contrlValue	Control's current setting
INTEGER	contrlMin	Control's minimum setting
INTEGER	contrlMax	Control's maximum setting
Handle	contrlDefProc	Control's definition function
Handle	contrlData	Data used by contrlDefProc
ProcPtr	contrlAction	Default action procedure
LONGINT	contrlRfCon	Control's reference value
Str255	contrlTitle	Control's title

Source: Inside Macintosh, page I-317

See Also: 3.083. Control Manager Routines

3.087. AuxCtlRec Data Structure

Size	*Name*	*Comments*
AuxCtlHandle	acNext	Handle to next record in list
ControlHandle	acOwner	Handle to owning control
CCTabHandle	acCTable	Handle to control's color table
INTEGER	acFlags	Miscellaneous flags (RESERVED)
LONGINT	acReserved	RESERVED
LONGINT	acRefCon	For application use

Source: Inside Macintosh, pages V-216 to V-217

See Also: 3.036. Color Table Data Structure
3.083. Control Manager Routines

3.088. CtlCTab Data Structure

Size	*Name*	*Comments*
LONGINT	ccSeed	NOT USED for controls
INTEGER	ccReserved	NOT USED for controls
INTEGER	ctSize	Number of entries in table minus 1
cSpecArray	ctTable	Array of ColorSpec records

Source: Inside Macintosh, page V-218

See Also: 3.037. ColorSpec Data Structure
3.083. Control Manager Routines

3.089. Control Part Identifiers for Color Tables

Value	*Constant*	*Comments*
0	cFrameColor	Color of control's frame
1	cBodyColor	Color of control's body
2	cTextColor	Color of control's text
3	cThumbColor	Color of control's thumb

Source: Inside Macintosh, page V-223

See Also: 3.083. Control Manager Routines

3.090. Control Axis Parameter Values

Value	*Constant*	*Comments*
0	noConstraint	No constraints on dragging control
1	hAxisOnly	Horizontal drag only
2	vAxisOnly	Vertical drag only

Source: Inside Macintosh, page I-325

See Also: 3.083. Control Manager Routines

3.091. Control Definition Messages

Value	*Constant*	*Comments*
0	drawCntl	Draw the control or relavent parts of it
1	testCntl	Test where the mouse was when button pushed
2	calcCRgns	Calculate control's region
3	initCntl	Perform additional control initialization
4	dispCntl	Perform additional control disposal actions
5	posCntl	Reposition control's indicator and update control
6	thumbCntl	Calculate parameters for dragging control's indicator
7	dragCntl	Drag control or its indicator
8	autoTrack	Execute control's action procedure

Source: Inside Macintosh, page I-329

See Also: 3.083. Control Manager Routines

3.092. Menu Manager Routines

Name	*Description*	*Defined* *	*Pascal Definition*
AddResMenu	Adds item(s) from resource file to end of menu	I	PROCEDURE AddResMenu (theMenu:MenuHandle; theType:ResType);
AppendMenu	Adds item(s) to end of menu	I	PROCEDURE AppendMenu (theMenu:MenuHandle; data:Str255);
CalcMenuSize	Recalculates h,v dimensions of menu (called internally)	I	PROCEDURE CalcMenuSize (theMenu:MenuHandle);
CheckItem	Places/removes check for menu item	I	PROCEDURE CheckItem (theMenu:MenuHandle; item:INTEGER; checked:BOOLEAN);
ClearMenuBar	Removes all menus from menu list	I	PROCEDURE ClearMenuBar;
CountMItems	Returns number of items in a menu	I	FUNCTION CountMItems (theMenu:MenuHandle) : INTEGER;
DeleteMenu	Removes menu from menu list	I	PROCEDURE DeleteMenu (menuID:INTEGER);
DelMCEntries	Removes entries from menu color info table	V	PROCEDURE DelMCEntries (MenuID,menuItem:INTEGER);
DelMenuItem	Removes menu item from menu	IV	PROCEDURE DelMenuItem (theMenu:MenuHandle; item:INTEGER);
DisableItem	Disables (makes gray) menu item or menu (if item = 0)	I	PROCEDURE DisableItem (theMenu:MenuHandle; item:INTEGER);
DispMCInfo	Releases memory and deallocates menu color info table	V	PROCEDURE DispMCInfo (menuCTbl:MCTableHandle);
DisposeMenu	Releases memory used by menu, deallocates it	I	PROCEDURE DisposeMenu (theMenu:MenuHandle);
DrawMenuBar	Redraws menu bar according to menu list	I	PROCEDURE DrawMenuBar;
EnableItem	Enables disabled menu item or menu (if item = 0)	I	PROCEDURE EnableItem (theMenu:MenuHandle; item:INTEGER);
FlashMenuBar	Inverts menu bar if menu isn't in current menu bar	I	PROCEDURE FlashMenuBar (menuID:INTEGER);
GetItem	Returns text for a menu item	I	PROCEDURE GetItem (theMenu:MenuHandle; item:INTEGER; VAR itemString:Str255);
GetItemCmd	Determines whether menu item has submenu	V	PROCEDURE GetItemCmd (theMenu:menuHandle; item:INTEGER; VAR cmdChar:Char);
GetItemIcon	Returns icon number associated with menu, if any	I	PROCEDURE GetItemIcon (theMenu:MenuHandle; item:INTEGER; VAR icon:Byte);
GetItemMark	Returns mark character, if any, for menu item	I	PROCEDURE GetItemMark (theMenu:MenuHandle; item:INTEGER; VAR markChar:CHAR);
GetItemStyle	Returns character style for menu item	I	PROCEDURE GetItemStyle (theMenu:MenuHandle; item:INTEGER; VAR chStyle:Style);
GetMCEntry	Returns pointer to entry in menu color info table	V	FUNCTION GetMCEntry (menuID,menuItem:INTEGER) : MCEntryPtr;
GetMCInfo	Returns handle to copy of current menu color info table	V	FUNCTION GetMCInfo : MCTableHandle;
GetMenu	Returns handle to menu with resource ID	I	FUNCTION GetMenu (resourceID:INTEGER) : MenuHandle;
GetMenuBar	Returns handle to copy of current menu list	I	FUNCTION GetMenuBar : Handle;
GetMHandle	Returns handle to menu in current menu bar	I	FUNCTION GetMHandle (menuID:INTEGER) : MenuHandle;
GetNewMBar	Returns handle to new menu list from MBAR resource	I	FUNCTION GetNewMBar (menuBarID:INTEGER) : Handle;
HiliteMenu	Highlights title of menu, if not already highlighted	I	PROCEDURE HiliteMenu (menuID:INTEGER);
InitMenus	Initializes the menu manager	I	PROCEDURE InitMenus;
InitProcMenu	Allocates new menu list for custom menu bar defproc	V	PROCEDURE InitProcMenu (mbResID:INTEGER);
InsertMenu	Adds menu to menu list before another menu	I	PROCEDURE InsertMenu (theMenu:MenuHandle; beforeID:INTEGER);
InsertResMenu	Adds item(s) from resource file to middle of menu	I	PROCEDURE InsertResMenu (theMenu:MenuHandle; theType:ResType; afterItem:INTEGER);
InsMenuItem	Inserts item(s) into a menu	IV	PROCEDURE InsMenuItem (theMenu:MenuHandle; itemString:Str255; afterItem:INTEGER);
InsNewItem	Inserts new menu item(s)	V	PROCEDURE InsNewItem (theMenu:MenuHandle; itemString:Str255; afterItem:INTEGER);
MenuChoice	Called after MenuSelect = 0 to find if disabled item was selected	V	FUNCTION MenuChoice : LONGINT;
MenuKey	Returns menu selection from command-key event	I	FUNCTION MenuKey (ch:CHAR) : LONGINT;
MenuSelect	Returns menu selection when mouse-down in menu bar	I	FUNCTION MenuSelect (startPt:Point) : LONGINT;
NewMenu	Returns handle to newly allocated menu	I	FUNCTION NewMenu (menuID:INTEGER; menuTitle:Str255) : MenuHandle;
PopUpMenuSelect	Creates popup menu	V	FUNCTION PopUpMenuSelect (theMenu:menuHandle; Top,Left,PopUpItem:INTEGER) : LONGINT;
SetItem	Changes text for menu item	I	PROCEDURE SetItem (theMenu:MenuHandle; item:INTEGER; itemString:Str255);

(continued)

Table 3.092. Continued

Name	*Description*	*Defined **	*Pascal Definition*
SetItemCmd	Attaches submenu to menu selection	V	PROCEDURE SetItemCmd (theMenu:menuHandle; item:INTEGER; cmdChar:Char);
SetItemIcon	Associates an icon with menu item	I	PROCEDURE SetItemIcon (theMenu:MenuHandle; item:INTEGER; icon:Byte);
SetItemMark	Places/removes mark for menu item	I	PROCEDURE SetItemMark (theMenu:MenuHandle; item:INTEGER; markChar:CHAR);
SetItemStyle	Changes character style for menu item	I	PROCEDURE SetItemStyle (theMenu:MenuHandle; item:INTEGER; chStyle:Style);
SetMCEntries	Adds entry(ies) to menu color info table	V	PROCEDURE SetMCEntries (numEntries:INTEGER; menuCEntries:MCTablePtr);
SetMCInfo	Copies menu color info table to current menu color info table	V	PROCEDURE SetMCInfo (menuCTbl:MCTableHandle);
SetMenuBar	Makes menu list the current one	I	PROCEDURE SetMenuBar (menuList:Handle);
SetMenuFlash	Sets blinking duration for menus selected	I	PROCEDURE SetMenuFlash (count:INTEGER);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-351 to I-361, IV-55 to IV-56, V-238 to V-247

See Also:
3.008. Interface Toolbox Managers
3.093. MenuInfo Data Structure
3.094. MCEntry Data Structure
3.095. Menu Color Information Table
3.096. MenuCRsrc Data Structure
3.097. Menu List Data Structure
3.098. MenuSelect ID/Item Return
3.099. Menu Definition Messages
3.100. Menu Definition Metacharacters
3.101. Menu Bar defProc Messages

3.093. MenuInfo Data Structure

Size	*Name*	*Comments*
INTEGER	menuID	ID for menu
INTEGER	menuWidth	Width of menu, in pixels
INTEGER	menuHeight	Height of menu, in pixels
Handle	menuProc	Menu definition procedure
LONGINT	enableFlags	Flags indicated enabled/disabled status *
varies	menuData **	Menu title string
varies	itemData **	Array of menu list data ***
Byte	endMarker	Should be zero

Notes: •* 1 = enabled, 0 = disabled; bit 0 is for menu itself, bits 1 to 31 are menu items 1 to 31, respectively.
•** Volume V of Inside Macintosh clarifies this structure rather explicitly; Volume I doesn't define it explicitly. Applications should not change this structure directly.
•*** itemString (string), itemIcon (byte), itemCmd (Char), itemMark (Char), itemStyle (Style).

Source: Inside Macintosh, pages I-345, V-230

See Also: 3.092. Menu Manager Routines

3.094. MCEntry Data Structure

Size	*Name*	*Values/Comments*
INTEGER	mctID	Menu ID (0 = menu bar)
INTEGER	mctItem	Menu item number (0 = menu title)
RGBColor	mctRGB1	See 3.095.
RGBColor	mctRGB2	See 3.095.
RGBColor	mctRGB3	See 3.095.
RGBColor	mctRGB4	See 3.095.
INTEGER	mctReserved	USED INTERNALLY

Source: Inside Macintosh, pages V-231 to V-234

See Also: 3.092. Menu Manager Routines
3.095. Menu Color Information Table

3.095. Menu Color Information Table

The RGB1, RGB2, RGB3, RGB4, and RESERVED fields in an MCEntry structure have differing meanings based on the values of the mctID and mctItem fields and on whether or not the MCEntry is for a menu bar, title, or item. This table outlines those differences.

mctID	0	<>0	<>0	-99
mctItem	0	0	<>0	RESERVED

	Menu Bar	*Menu Title*	*Menu Item*	*Last Entry*
RGB1	Default color for menu titles	Title color	Mark color	RESERVED
RGB2	Default color for pulldown menu	Menu bar color	Name color	RESERVED
RGB3	Default color for items in menu	Default color for items in menu	Command key equivalent color	RESERVED
RGB4	Menu bar color	Menu background color	Menu background color	RESERVED
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED

Notes: This is a complex table; read the source for additional insight.

Source: Inside Macintosh, pages V-233 to V-234

See Also: 3.092. Menu Manager Routines
3.094. MCEntry Data Structure

3.096. MenuCRsrc Data Structure

Size	*Name*	*Comments*
INTEGER	numEntries	Number of MCEntry structures in structure
MCEntry array	data	See 3.094.

Source: Inside Macintosh, page V-234

See Also: 3.092. Menu Manager Routines
3.094. MCEntry Data Structure
3.095. Menu Color Information Table

3.097. Menu List Data Structure

Size	*Name*	*Comments*
Word	itemOffset	Offset to last menu handle in structure (numItems*6)
Word	hCoord	Horizontal coordinate of right edge of menu title for last menu
Word	NOT USED	NOT USED
Varies	menuData	For each menu: menuHandle, Pt (h coord of left edge of menu)

Notes: Generally only of use to assembly-language programmers.

Source: Inside Macintosh, page I-346

See Also: 3.092. Menu Manager Routines

3.098. MenuSelect ID/Item Return

Size	*Name*	*Comments*
HO word	menuSelect	Menu ID of selected item
LO word	menuKey	Item number of selected item

Source: Inside Macintosh, page I-356

See Also: 3.092. Menu Manager Routines

3.099. Menu Definition Messages

Value	*Constant*	*Comments*
0	mDrawMsg	Draw the menu
1	mChooseMsg	Indicate choosen item and highlight it
2	mSizeMsg	Calculate menu's dimension

Source: Inside Macintosh, pages I-362 to I-363

See Also: 3.092. Menu Manager Routines

3.100. Menu Definition Metacharacters

Character	*Meaning*	*Example*
;	Item separator *	'Cut';'Copy'
^	Include icon with item **	'Cut^1'
!	Include checkmark with item	'Cut!'
<	Format item with style ***	'Cut<B'
/	Include keyboard equivalent with item	'Cut/X'
(	Disable item	'(Cut'

Notes:
- * Character can also be RETURN.
- ** Icon ID follows ^.
- *** <B = bold, <I = italic, <U = underlined, <O = outlined, <S = shadow.

Source: Inside Macintosh, pages I-346 to I-349

See Also: 3.092. Menu Manager Routines

3.101. Menu Bar defProc Messages

Value	*Constant*	*Comments*
0	Draw	Draws/clears menu bar
1	Hit	Tests mouse for menu bar, menu hit
2	Calc	Calculates left edges of menu title in menuList
3	Init	Initializes menu bar defProc structures
4	Dispos	Releases menu bar defProc structures
5	Hilite	Highlights menu title or inverts menu bar
6	Height	Returns menu bar height
7	Save	Saves bits behind menu, draws menu
8	Restor	Restores bits behind menu
9	Rect	Calculates menu rectangle
10	SaveAlt	Saves information about menu after it is drawn
11	ResetAlt	Resets information about menu
12	MenuRgn	Returns region occupied by menu bar

Source: Inside Macintosh, pages V-250 to V-251

See Also: 3.092. Menu Manager Routines

3.102. TextEdit Routines

Name	*Description*	*Defined **	*Pascal Definition*
GetStylHandle	Returns edit record's style handle	V	FUNCTION GetStylHandle (hTE:TEHandle) : TEStyleHandle;
GetStylScrap	Copies selected text to style scrap	V	FUNCTION GetStylScrap (hTE:TEHandle) : StScrpHandle;
SetClikLoop	Installs click loop procedure	I	PROCEDURE SetClikLoop (clikProc:ProcPtr; hTE:TEHandle);
SetStylHandle	Sets edit record's style handle	V	PROCEDURE SetStylHandle (theHandle:TEStyleHandle; hTE:TEHandle);
SetWordBreak	Installs word break procedure	I	PROCEDURE SetWordBreak (wBrkProc:ProcPtr; hTE:TEHandle);
TEActivate	Highlights selected range	I	PROCEDURE TEActivate (hTE:TEHandle);
TEAutoView	Enables/disables automatic scrolling	IV	PROCEDURE TEAutoView (auto:BOOLEAN; hTE:TEHandle);
TECalText	Recalculates beginnings of all lines of text	I	PROCEDURE TECalText (hTE:TEHandle);
TEClick	Handles selection/highlight after mouse-down event	I	PROCEDURE TEClick (pt:Point; extend:BOOLEAN; hTE:TEHandle);
TECopy	Copies selected range to scrap	I	PROCEDURE TECopy (hTE:TEHandle);
TECut	Removes selection range from text, puts in scrap	I	PROCEDURE TECut (hTE:TEHandle);
TEDeactivate	Unhighlights selected range of text	I	PROCEDURE TEDeactivate (hTE:TEHandle);
TEDelete	Removes selected range, redraws text	I	PROCEDURE TEDelete (hTE:TEHandle);
TEDispose	Releases memory used by edit record, deallocates	I	PROCEDURE TEDispose (hTE:TEHandle);
TEFromScrap	Copies desk scrap to TE scrap	I	FUNCTION TEFromScrap : OSErr;
TEGetHeight	Returns height of all lines of text, in pixels	V	FUNCTION TEGetHeight (endLine,startLine:LONGINT; hTE:TEHandle) : INTEGER;
TEGetOffset	Finds character offset in edit record corresponding to point	V	FUNCTION TEGetOffset (pt:Point; hTE:TEHandle) : INTEGER;
TEGetPoint	Returns point corresponding to given offset into text	V	FUNCTION TEGetPoint (offset:INTEGER; hTE:TEHandle) : POINT;
TEGetScrapLen	Returns size of TE scrap, in bytes	I	FUNCTION TEGetScrapLen : LONGINT;
TEGetStyle	Returns style information	V	PROCEDURE TEGetStyle (offset:INTEGER; VAR theStyle:TextStyle; VAR lineHeight,fontAscent:INTEGER; hTE:TEHandle);
TEGetText	Returns handle to text for edit record	I	FUNCTION TEGetText (hTE:TEHandle) : CharsHandle;
TEIdle	Makes blinking caret appear at insertion point	I	PROCEDURE TEIdle (hTE:TEHandle);
TEInit	Initializes TextEdit	I	PROCEDURE TEInit;
TEInsert	Inserts text before selected range	I	PROCEDURE TEInsert (text:Ptr; length:LONGINT; hTE:TEHandle);
TEKey	Replaces selected range with character	I	PROCEDURE TEKey (key:CHAR; hTE:TEHandle);
TENew	Returns handle to new edit record	I	FUNCTION TENew (destRect,viewRect:Rect) : TEHandle;
TEPaste	Replaces selected range with scrap	I	PROCEDURE TEPaste (hTE:TEHandle);
TEPinScroll	Similar to TEScroll, but stops scrolling at last line	IV	PROCEDURE TEPinScroll (dh,dv:INTEGER; hTE:TEHandle);
TEReplaceStyle	Replaces style of current selection	V	PROCEDURE TEReplaceStyle (mode:INTEGER; oldStyle,newStyle:TextStyle; redraw:BOOLEAN; hTE:TEHandle);
TEScrapHandle	Returns handle to TE scrap	I	FUNCTION TEScrapHandle : Handle;
TEScroll	Scrolls text within view rectangle by dh,dv pixels	I	PROCEDURE TEScroll (dh,dv:INTEGER; hTE:TEHandle);
TESelView	Ensures selection range is visible in autoscroll	IV	PROCEDURE TESelView (hTE:TEHandle);
TESetJust	Sets justification type of text	I	PROCEDURE TESetJust (just:INTEGER; hTE:TEHandle);
TESetScrapLen	Sets size of TE scrap, in bytes	I	PROCEDURE TESetScrapLen (length:LONGINT);
TESetSelect	Sets selection range in text	I	PROCEDURE TESetSelect (selStart,selEnd:LONGINT; hTE:TEHandle);
TESetStyle	Sets style of current selection	V	PROCEDURE TESetStyle (mode:INTEGER; newStyle:TextStyle; redraw:BOOLEAN; hTE:TEHandle);
TESetText	Changes edit record to new text	I	PROCEDURE TESetText (text:Ptr; length:LONGINT; hTE:TEHandle);
TEStylInsert	Inserts styled text before hTE	V	PROCEDURE TEStylInsert (text:Ptr; length:LONGINT; hST:stScrpHandle; hTE:TEHandle);
TEStylNew	Creates new style edit record	V	FUNCTION TEStyleNew (destRect,viewRect:Rect) : TEHandle;
TEStylPaste	Pastes text from desk scrap to current insertion point	V	PROCEDURE TEStylPaste (hTE:TEHandle);
TEToScrap	Copies TE scrap to desk scrap	I	FUNCTION TEToScrap : OSErr;
TEUpdate	Draws hTE text within rUpdate rectangle	I	PROCEDURE TEUpdate (rUpdate:Rect; hTE:TEHandle);
TextBox	Draws text in rectangle with justification	I	PROCEDURE TextBox (text:Ptr; length:LONGINT; box:Rect; just:INTEGER);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-383 to I-391, IV-57, V-267 to V-270

See Also: 3.008. Interface Toolbox Managers

3.103. TERec Data Structure

Size	*Name*	*Comments*
Rect	destRect	Destination rectangle
Rect	viewRect	Viewing rectangle
Rect	selRect	Used by assembly-language routines only
INTEGER	lineHeight	Line spacing *
INTEGER	fontAscent	Caret/hilite position *
Point	selPoint	Point selected with mouse
INTEGER	selStart	Start of selection range
INTEGER	selEnd	End of selection range
INTEGER	active	USED INTERNALLY
ProcPtr	wordBreak	Word break routine, if any
ProcPtr	clikLoop	Click loop routine, if any
LONGINT	clickTime	USED INTERNALLY
INTEGER	clickLoc	USED INTERNALLY
LONGINT	caretTime	USED INTERNALLY
INTEGER	caretState	USED INTERNALLY
INTEGER	just	Type of text justification
INTEGER	teLength	Length of text
Handle	hText	Text to be edited
INTEGER	recalBack	USED INTERNALLY
INTEGER	recalLines	USED INTERNALLY
INTEGER	clikStuff	USED INTERNALLY
INTEGER	crOnly	If <0, new line at Return only
INTEGER	txFont	Text font
Style	txFace	Character style
INTEGER	txMode	Pen mode
INTEGER	txSize	Font size **
GrafPtr	inPort	GrafPort
ProcPtr	highHook	Text highlighting routine
ProcPtr	caretHook	Caret display routine
INTEGER	nLines	Number of lines in array
Varies	lineStarts	INTEGER array of positions of line starts

Notes: •* -1 = calculated for each line; >0 fixed line height or font ascent.
•** -1 = style information available, txFont/txFace hold handle to TEStyleRec; ≥0 = old-style record.

Source: Inside Macintosh, pages I-377 to I-381, V-260 to V-261

See Also: 3.102. TextEdit Routines

3.104. TextEdit Justification Codes

Value	*Constant*	*Comments*
-1	teJustRight	Right justified
0	teJustCenter	Centered
1	teJustLeft	Left justified (default)

Source: Inside Macintosh, page I-387

See Also: 3.102. TextEdit Routines
3.103. TERec Data Structure

3.105. TEStyleRec Data Structure

Size	*Name*	*Comments*
INTEGER	nRuns	Number of style runs
INTEGER	nStyles	Number of distinct styles
StHandle	styleTab	Style table
LHHandle	lhTab	Line height table
LONGINT	teRefCon	RESERVED for application use
nullStHandle	nullStyle	Style set at null selection
varies	runs	Array of StyleRuns *

Notes: •* Startchar (INTEGER), styleIndex (INTEGER).

Source: Inside Macintosh, pages V-261 to V-262

See Also: 3.102. TextEdit Routines
3.106. STElement/LHElement Data Structures

3.106. STElement/LHElement Data Structures

STElement

Size	*Name*	*Comments*
INTEGER	stCount	Number of runs in style
INTEGER	stHeight	Line height
INTEGER	stAscent	Font ascent
INTEGER	stFont	Font family number
Style	stFace	Character style
INTEGER	stSize	Size in points
RGBColor	stColor	Absolute RGBColor

LHElement

Size	*Name*	*Comments*
INTEGER	lhHeight	Maximum height in line
INTEGER	lhAscent	Maximum ascent in line

Source: Inside Macintosh, pages V-262 to V-264

See Also: 3.035. RGBColor Data Structure
3.102. TextEdit Routines
3.105. TEStyleRec Data Structure

3.107. NullSTRec Data Structure

Size	*Name*	*Comments*
LONGINT	TEReserved	RESERVED
STScrpHandle	nullScrap	Scrap style table

Source: Inside Macintosh, page V-264

See Also: 3.102. TextEdit Routines

3.108. TextStyle Data Structure

Size	*Name*	*Comments*
INTEGER	tsFont	Font family number
Style	tsFace	Character style
INTEGER	tsSize	Size of font, in points
RGBColor	tsColor	Absolute RGB color

Source: Inside Macintosh, page V-265

See Also: 3.035. RGBColor Data Structure
3.102. TextEdit Routines

3.109. StScrpRec Data Structure

Size	*Name*	*Comments*
INTEGER	scrpNStyles	Number of distinct styles in scrap
ScrpSTTable	scrpStyleTab	Table of scrap styles

Source: Inside Macintosh, page V-265

See Also: 3.102. TextEdit Routines

3.110. ScrpSTElement Data Structure

Size	*Name*	*Comments*
LONGINT	scrpStartChar	Offset to start of style
INTEGER	scrpHeight	Line height
INTEGER	scrpAscent	Font ascent
INTEGER	scrpFont	Font family number
Style	scrpFace	Character style
INTEGER	scrpSize	Size in points
RGBColor	scrpColor	Absolute RGB color

Source: Inside Macintosh, page V-266

See Also: 3.035. RGBColor Data Structure
3.102. TextEdit Routines

3.111. SetStyle Attribute Codes

Value	*Constant*	*Comments*
1	doFont	Set font family number
2	doFace	Set character style
4	doSize	Set font size
8	doColor	Set font color
15	doAll	Set all attributes for font
16	addSize	Adjust font size

Source: Inside Macintosh, page V-269

See Also: 3.102. TextEdit Routines

3.112. Dialog Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
Alert	Displays and handles alert box	I	FUNCTION Alert (alertID:INTEGER; filterProc:ProcPtr) : INTEGER;
CautionAlert	Displays and handles caution alert box	I	FUNCTION CautionAlert (alertID:INTEGER; filterProc:ProcPtr) : INTEGER;
CloseDialog	Removes dialog from window and dialog list	I	PROCEDURE CloseDialog (theDialog:DialogPtr);
CouldAlert	Makes alert template unpurgeable	I	PROCEDURE CouldAlert (alertID:INTEGER);
CouldDialog	Makes dialog template unpurgeable	I	PROCEDURE CouldDialog (dialogID:INTEGER);
DialogSelect	Handles events in modeless dialog	I	FUNCTION DialogSelect (theEvent:EventRecord; VAR theDialog:DialogPtr; VAR itemHit:INTEGER) : BOOLEAN;
DisposDialog	Releases memory for dialog	I	PROCEDURE DisposDialog (theDialog:DialogPtr);
DlgCopy	Applies TECopy to selected editText item	I	PROCEDURE DlgCopy (theDialog:DialogPtr);
DlgCut	Applies TECut to selected editText item	I	PROCEDURE DlgCut (theDialog:DialogPtr);
DlgDelete	Applies TEDelete to selected editText item	I	PROCEDURE DlgDelete (theDialog:DialogPtr);
DlgPaste	Applies TEPaste to selected editText item	I	PROCEDURE DlgPaste (theDialog:DialogPtr);
DrawDialog	Draws contents of dialog box	I	PROCEDURE DrawDialog (theDialog:DialogPtr);
ErrorSound	Sets sound procedure for alerts	I	PROCEDURE ErrorSound (soundProc:ProcPtr);
FindDItem	Returns item number that contains point	IV	FUNCTION FindDItem (theDialog:DialogPtr; thePt:Point) : INTEGER;
FreeAlert	Makes alert template purgeable	I	PROCEDURE FreeAlert (alertID:INTEGER);
FreeDialog	Makes dialog template purgeable	I	PROCEDURE FreeDialog (dialogID:INTEGER);
GetAlrtStage	Returns stage of last alert	I	FUNCTION GetAlrtStage : INTEGER;
GetDItem	Returns information about dialog item	I	PROCEDURE GetDItem (theDialog:DialogPtr; itemNo:INTEGER; VAR itemType:INTEGER; VAR item:Handle; VAR box:Rect);
GetIText	Returns text of statText or editText item	I	PROCEDURE GetIText (item:Handle; VAR text:Str255);
GetNewDialog	Returns handle to dialog created with resource	I	FUNCTION GetNewDialog (dialogID:INTEGER; dStorage:Ptr; behind:WindowPtr) : DialogPtr;
HideDItem	Hides dialog item by moving it offscreen	IV	PROCEDURE HideDItem (theDialog:DialogPtr; itemNo:INTEGER);
InitDialogs	Initializes dialog manager	I	PROCEDURE InitDialogs (resumeProc:ProcPtr);
IsDialogEvent	Returns TRUE if modeless dialog event	I	FUNCTION IsDialogEvent (theEvent:EventRecord) : BOOLEAN;
ModalDialog	Gets and handles events in dialog window	I	PROCEDURE ModalDialog (filterProc:ProcPtr; VAR itemHit:INTEGER);
NewCDialog	Returns handle to new color dialog	V	FUNCTION NewCDialog (dStorage:Ptr; boundsRect:Rect; title:Str255; visible:BOOLEAN; procID:INTEGER; behind:WindowPtr;goAwayFlag:BOOLEAN; refCon:LONGINT; items:Handle) : CDialogPtr;
NewDialog	Returns handle to newly created dialog	I	FUNCTION NewDialog (dStorage:Ptr; boundsRect:Rect; title:Str255; visible:BOOLEAN; procID:INTEGER; behind:WindowPtr; goAwayFlag:BOOLEAN; refCon:LONGINT; items:Handle) : DialogPtr;
NoteAlert	Displays and handles note alert box	I	FUNCTION NoteAlert (alertID:INTEGER; filterProc:ProcPtr) : INTEGER;
ParamText	Substitutes text for parameters (^0 to ^3)	I	PROCEDURE ParamText (param0,param1,param2,param3:Str255);
ResetAlrtStage	Resets alert stages so next alert will be first stage	I	PROCEDURE ResetAlrtStage;
SelIText	Sets selection range for editText	I	PROCEDURE SelIText (theDialog:DialogPtr; itemNo:INTEGER; strtSel,endSel:INTEGER);
SetDAFont	Sets font for dialogs or alerts	I	PROCEDURE SetDAFont (fontNum:INTEGER);
SetDItem	Sets information about dialog item	I	PROCEDURE SetDItem (theDialog:DialogPtr;itemNo:INTEGER; itemType:INTEGER; item:Handle; box:Rect);
SetIText	Sets text for statText or editText item	I	PROCEDURE SetIText (item:Handle; text:Str255);
ShowDItem	Redisplays dialog item	IV	PROCEDURE ShowDItem (theDialog:DialogPtr; itemNo:INTEGER);
StopAlert	Displays and handles stop alert box	I	FUNCTION StopAlert (alertID:INTEGER; filterProc:ProcPtr) : INTEGER;
UpdtDialog	Faster DrawDialog function	IV	PROCEDURE UpdtDialog (theDialog:DialogPtr; updateRgn:RgnHandle);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-411 to I-423, IV-59 to IV-60, V-283 to V-285

See Also:
3.008. Interface Toolbox Managers
3.113. Dialog Item Types
3.114. Dialog Item Lists
3.115. Dialog Color Table
3.116. Dialog Standard Item Constants
3.117. DialogRecord Data Structure
3.118. DialogTemplate Data Structure
3.119. AlertTemplate Data Structure
3.120. Alert StageList Data Structure

3.113. Dialog Item Types

Type	*Description*
ctrlItem + btnCtrl	Standard button control
ctrlItem + chkCtrl	Standard check box control
ctrlItem + radCtrl	Standard radio button control
ctrlItem + resCtrl	Control as defined in control template of resource file
statText	Static text
editText	Editable text (in dialogs only)
iconItem	Icon
picItem	QuickDraw picture
userItem	Application-defined item
itemDisable + (any of above)	Disabled item *

Notes: * Dialog manager doesn't report events for disabled items.

Source: Inside Macintosh, page I-404

See Also: 3.112. Dialog Manager Routines
3.114. Dialog Item Lists

3.114. Dialog Item Lists

An item list is a list of item types, which contain data about the items:

Type	*Contains*
ctrlItem	Handle to the control
statText	Handle to the text
editText	Handle to the current text
iconItem	Handle to the icon
picItem	Handle to the picture
userItem	Pointer to the procedure

Source: Inside Macintosh, page I-405

See Also: 3.112. Dialog Manager Routines
3.113. Dialog Item Types

3.115. Dialog Color Table

Size	*Name*	*Values/Comments*
Long word	wCSeed	Always 0
Word	wCReserved	Always 0
Word	ctSize	-1 for default window colors
Word	wContentColor	Part ID for content
RGBColor	wContentRGB	Content's color
Word	wFrameColor	Part ID for frame
RGBColor	wFrameRGB	Frame's color
Word	wTextColor	Part ID for text
RGBColor	wTextRGB	Text's color
Word	wHiliteColor	Part ID for hilite
RGBColor	wHiliteRGB	Highlights color
Word	wTitleBarColor	Part ID for title bar
RGBColor	wTitleBarRGB	Title bar's color

Source: Inside Macintosh, pages V-278 to V-279

See Also: 3.112 Dialog Manager Routines

3.116. Dialog Standard Item Constants

Value	*Constant*	*Comments*
1	OK	Perform action
2	cancel	Cancel action – do nothing

Source: Inside Macintosh, page I-407

See Also: 3.112. Dialog Manager Routines

3.117. DialogRecord Data Structure

Size	*Name*	*Values/Comments*
WindowRecord	window	Dialog window information
Handle	items	Dialog item list
TEHandle	textH	Current editText item, if any
INTEGER	editField	EditText item number minus 1
INTEGER	editOpen	USED INTERNALLY
INTEGER	aDefItem	Default button item number

Source: Inside Macintosh, page I-408

See Also: 3.075. WindowRecord Data Structure
3.112. Dialog Manager Routines

3.118. DialogTemplate Data Structure

Size	*Name*	*Values/Comments*
Rect	boundsRect	Window's portRect
INTEGER	procID	Window def ID
BOOLEAN	visible	TRUE = visible
BOOLEAN	filler1	NOT USED
BOOLEAN	goAwayFlag	TRUE = has goaway region
BOOLEAN	filler2	NOT USED
LONGINT	refCon	Window's reference value
INTEGER	itemsID	Resource ID of item list
Str255	title	Window's title

Source: Inside Macintosh, pages I-423 to I-424

See Also: 3.112. Dialog Manager Routines

3.119. AlertTemplate Data Structure

Size	*Name*	*Values/Comments*
Rect	boundsRect	Window's portRect
INTEGER	itemsID	Resource ID of item list
StageList	stages	See 3.120.

Source: Inside Macintosh, page I-424

See Also: 3.112. Dialog Manager Routines
3.120. Alert StageList Data Structure

3.120. Alert StageList Data Structure

Bit Numbers

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Name*	*Description*
X																boldItm4	Default button item number minus 1
	X															boxDrwn4	TRUE = draw box
		X	X													sound4	Sound number for stage
				X												boldItm3	Default button item number minus 1
					X											boxDrwn3	TRUE = draw box
						X	X									Sound3	Sound number for stage
								X								boldItm2	Default button item number minus 1
									X							boxDrwn2	TRUE = draw box
										X	X					sound2	Sound number for stage
												X				boldItm1	Default button item number minus 1
													X			boxDrwn1	TRUE = draw box
														X	X	sound1	Sound number for stage

Source: Inside Macintosh, pages I-424 to I-425

See Also: 3.112. Dialog Manager Routines
3.119. AlertTemplate Data Structure

3.121. Desk Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
OpenDeskAcc	Opens named DA and displays its window	I	FUNCTION OpenDeskAcc (theAcc:Str255) : INTEGER;
CloseDeskAcc	Closes desk accessory	I	PROCEDURE CloseDeskAcc (refNum:INTEGER);
SystemClick	Handles DA window events	I	PROCEDURE SystemClick (theEvent:EventRecord; theWindow:WindowPtr);
SystemEdit	Handles DA use of clipboard commands	I	FUNCTION SystemEdit (editCmd:INTEGER) : BOOLEAN;
SystemTask	Handles systematic action for DA, if any	I	PROCEDURE SystemTask;
SystemEvent	Called by Event Manager when it receives an event	I	FUNCTION SystemEvent (theEvent:EventRecord) : BOOLEAN;
SystemMenu	Called by Menu Manager when DA menu item chosen	I	PROCEDURE SystemMenu (menuResult:LONGINT);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-440 to I-443

See Also: 3.008. Interface Toolbox Managers
3.122. editCmd Parameters
3.123. DA Device Driver Format
3.124. DA Device Driver Control Routine Messages

3.122. editCmd Parameters

Value	*Constant*
0	Undo
2	Cut
3	Copy
4	Paste
5	Clear

Source: Inside Macintosh, pages I-441 to I-442

See Also: 3.121. Desk Manager Routines

3.123. DA Device Driver Format

Size	*Name*	*Values/Comments*
Word	drvrFlags	See source
Word	drvrDelay	Time between actions, in ticks
Word	drvrEMask	Event mask
Word	drvrMenu	Menu ID associated with driver
Word	drvrOpen	Offset to open routine
Word	drvrPrime	Offset to prime routine
Word	drvrCtl	Offset to control routine
Word	drvrStatus	Offset to status routine
Word	drvrClose	Offset to close routine
Byte	drvrName	Length of name
Varies	drvrName + 1	Driver name
Varies	driver routines	

Source: Inside Macintosh, pages I-443 to I-447

See Also: 3.121. Desk Manager Routines
3.124. DA Device Driver Control Routine Messages

3.124. DA Device Driver Control Routine Messages

Value	*Constant*	*Comments*
64	accEvent	Handle the event
65	accRun	Take periodic action, if any
66	accCursor	Change cursor shape, if appropriate
67	accMenu	Handle menu item
68	accUndo	Handle Undo command
70	accCut	Handle Cut command
71	accCopy	Handle Copy command
72	accPaste	Handle Paste command
73	accClear	Handle Clear command

Source: Inside Macintosh, page I-446

See Also: 3.121. Desk Manager Routines
3.123. DA Device Driver Format

3.125. Scrap Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
GetScrap	Reads data from desk scrap to copy in memory	I	FUNCTION GetScrap (hDest:Handle; theType:ResType; VAR offset:LONGINT) : LONGINT;
InfoScrap	Returns pointer to ScrapStuff structure	I	FUNCTION InfoScrap : PScrapStuff;
LoadScrap	Reads scrap from file to memory	I	FUNCTION LoadScrap : LONGINT;
PutScrap	Writes data at source to desk scrap	I	FUNCTION PutScrap (length:LONGINT; theType:ResType; source:Ptr) : LONGINT;
UnloadScrap	Writes scrap from memory to file, releases memory	I	FUNCTION UnloadScrap : LONGINT;
ZeroScrap	Clears contents of existing scrap, or starts new one	I	FUNCTION ZeroScrap : LONGINT;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-457 to I-460

See Also: 3.008. Interface Toolbox Managers
3.126. ScrapStuff Data Structure
3.127. Scrap Data Formats
3.128. Scrap Manager Result Codes

3.126. ScrapStuff Data Structure

Size	*Name*	*Values/Comments*
LONGINT	scrapSize	Size of scrap, in bytes
Handle	scrapHandle	Handle to desk scrap
INTEGER	scrapCount	Changed each time ZeroScrap called *
INTEGER	scrapState	0 = on disk, >0 = in memory, <0 = uninitialized
StringPtr	scrapName	Usually "Clipboard File"

Notes: * Only tells you whether contents have been changed.

Source: Inside Macintosh, page I-457

See Also: 3.125. Scrap Manager Routines

3.127. Scrap Data Formats

Size	*Name*	*Values/Comments*
Dbl word	scrapType	Usually either 'TEXT' or 'PICT'; resource format
Dbl word	scrapLength	Length of scrap data, in bytes
Varies *	scrapData	**

Notes:
- * Must be an even number (pad with zero byte if not).
- ** Example: TEXT data format = Length (dbl word), text (varies).

Source: Inside Macintosh, page I-462

See Also: 3.125. Scrap Manager Routines

3.128. Scrap Manager Result Codes

Value	*Constant*	*Comments*
-100	noScrapErr	Desk scrap has not been initialized
-102	noTypeErr	No data of requested type was found

Source: Inside Macintosh, pages I-463 to I-464

See Also: 3.125. Scrap Manager Routines

3.129. Toolbox Utility Routines

Name	*Description*	*Defined **	*Pascal Definition*
AngleFromSlope	Returns angle calculated from slope dh/dv	I	FUNCTION AngleFromSlope (slope:Fixed) : INTEGER;
BitAnd	Returns result of bitwise AND operation	I	FUNCTION BitAnd (value1,value2:LONGINT) : LONGINT;
BitClr	Clears bit (makes it 0)	I	PROCEDURE BitClr (bytePtr:Ptr; bitNum:LONGINT);
BitNot	Returns result of bitwise NOT operation	I	FUNCTION BitNot (value:LONGINT) : LONGINT;
BitOr	Returns result of bitwise OR operation	I	FUNCTION BitOr (value1,value2:LONGINT) : LONGINT;
BitSet	Sets bit (makes it 1)	I	PROCEDURE BitSet (bytePtr:Ptr; bitNum:LONGINT);
BitShift	Returns result of bitwise shift, MOD 32	I	FUNCTION BitShift (value:LONGINT; count:INTEGER) : LONGINT;
BitTst	Returns TRUE if bit is set	I	FUNCTION BitTst (bytePtr:Ptr;bitNum:LONGINT) : BOOLEAN;
BitXor	Returns result of bitwise XOR operation	I	FUNCTION BitXor (value,value2:LONGINT) : LONGINT;
DeltaPoint	Subtracts ptB coords from ptA coords (HO result = v,LO = h)	I	FUNCTION DeltaPoint (ptA,ptB:Point) : LONGINT;
FixATan2	Returns arctangent of y/x, in radians **	IV	FUNCTION FixATan2 (x,y:LONGINT) : Fixed;
FixDiv	Returns x/y **	IV	FUNCTION FixDiv (x,y:Fixed) : Fixed;
FixMul	Returns fp product of a/b	I	FUNCTION FixMul (a,b:Fixed) : Fixed;
FixRatio	Returns fp quotient of numer/denom	I	FUNCTION FixRatio (numer,denom:INTEGER) : Fixed;
FixRound	Returns rounded integer of positive fp number	I	FUNCTION FixRound (x:Fixed) : INTEGER;
FracCos	Returns cosine of radian argument, x	IV	FUNCTION FracCos (x:Fixed) : Fract;
FracDiv	Returns x/y **	IV	FUNCTION FracDiv (x,y:Fract) : Fract;
FracMul	Returns x*y **	IV	FUNCTION FracMul (x,y:Fract) : Fract;
FracSin	Returns sine of radian argument, x	IV	FUNCTION FracSin (x:Fixed) : Fract;
FracSqrt	Returns square root of x	IV	FUNCTION FracSqrt (x:Fract) : Fract;
GetCursor	Returns handle to CURS in resource file	I	FUNCTION GetCursor (cursorID:INTEGER) : CursHandle;
GetIcon	Returns handle to ICON in resource file	I	FUNCTION GetIcon (iconID:INTEGER) : Handle;
GetIndPattern	Returns PAT# in resource file	I	PROCEDURE GetIndPattern (VAR thePattern:Pattern; patListID:INTEGER; index:INTEGER);
GetIndString	Returns handle to string in STR# resource	I	PROCEDURE GetIndString (VAR theString:Str255; strListID:INTEGER; index:INTEGER);
GetPattern	Returns handle to PAT in resource file	I	FUNCTION GetPattern (patID:INTEGER) : PatHandle;
GetPicture	Returns handle to PICT in resource file	I	FUNCTION GetPicture (picID:INTEGER) : PicHandle;
GetString	Returns handle to string with resource ID	I	FUNCTION GetString (stringID:INTEGER) : StringHandle;
HiWord	Returns HO word of integer	I	FUNCTION HiWord (x:LONGINT) : INTEGER;
LongMul	Returns Int64Bit product of a * b	I	PROCEDURE LongMul (a,b:LONGINT; VAR dest:Int64Bit);
LoWord	Returns LO word of integer	I	FUNCTION LoWord (x:LONGINT) : INTEGER;
Munger	Manipulates bytes of a string	I	FUNCTION Munger (h:Handle; offset:LONGINT; ptr1:Ptr; len1:LONGINT; ptr2:Ptr; len2:LONGINT) : LONGINT;
NewString	Returns handle to newly allocated string	I	FUNCTION NewString (theString:Str255) : StringHandle;
PackBits	Compresses data	I	PROCEDURE PackBits (VAR srcPtr,dstPtr:Ptr; srcBytes:INTEGER);
PlotIcon	Draws icon in rectangle	I	PROCEDURE PlotIcon (theRect:Rect; theIcon:Handle);
ScreenRes	Returns resolution of screen being used, in pixels per inch	I	PROCEDURE ScreenRes (VAR scrnHRes,scrnVRes:INTEGER);
SetString	Sets string to new value	I	PROCEDURE SetString (h:StringHandle; theString:Str255);
ShieldCursor	Hides cursor if in given rectangle	I	PROCEDURE ShieldCursor (shieldRect:Rect; offsetPt:Point);
SlopeFromAngle	Returns slope dh/dv calculated from angle	I	FUNCTION SlopeFromAngle (angle:INTEGER) : Fixed;
SysEnvirons	Returns pointer to SysEnvRecord	V	FUNCTION SysEnvrions (versRequested:INTEGER; VAR theWorld:SysEnvRecPtr) : OSErr;
UnpackBits	Decompresses data	I	PROCEDURE UnpackBits (VAR srcPtr,dstPtr:Ptr; dstBytes:INTEGER);

Notes:
- * Volume of Inside Macintosh where first definition of routine appeared (I–V).
- ** Note that result type changes depending on types in original values; see source. Fract data type: see IV-63.

Source: Inside Macintosh, pages I-467 to I-472, IV-64 to IV-65, V-6 to V-8

See Also:
3.008. Interface Toolbox Managers
3.130. FixedAndInt Data Structure
3.131. Int64Bit Data Structure
3.132. SysEnvRec Data Structure
3.133. SysEnv Result Codes

3.130. FixedAndInt Data Structure

If structure holds Fixed value:

Size	*Name*	*Values/Comments*
Fixed	fixedView	Standard fixed value

If structure holds Integer value:

Size	*Name*	*Values/Comments*
INTEGER	whole	Integer value
INTEGER	part	Fractional value

Source: Inside Macintosh, page I-472

See Also: 3.129. Toolbox Utility Routines

3.131. Int64Bit Data Structure

Size	*Name*	*Values/Comments*
LONGINT	hiLong	HO portion of result
LONGINT	loLong	LO portion of result

Source: Inside Macintosh, page I-472

See Also: 3.129. Toolbox Utility Routines

3.132. SysEnvRec Data Structure

Size	*Name*	*Description*	*Values*	*Constant*
INTEGER	environsVersion	SysEnv version number	Currently 1	
INTEGER	machineType	Type of Mac	0 = unknown 1 = 512KE 2 = Plus 3 = SE 4 = II -1 = 64K ROM -2 = Mac XL (Lisa)	envMachUnknown env512KE envMacPlus envSE envMacII envMac envXL
INTEGER	systemVersion	Version of System File		
INTEGER	processor	Processor type	0 = unknown 1 = 68000 2 = 68010 3 = 68020	envCPUUnknown env68000 env68010 env68020
BOOLEAN	hasFPU	Is 68881/68882 coprocessor available		
BOOLEAN	hasColorQD	Is color QuickDraw available		
INTEGER	keyBoardType	Type of keyboard being used	0 = Plus keyboard with keypad 1 = Mac keyboard 2 = Mac keyboard with keypad 3 = Plus keyboard 4 = Extended keyboard (ADB) 5 = Standard keyboard (ADB)	envUnknownKbd envMacKbd envMacAndPad envMacPlusKbd envAExtendKbd envStandADBKbd
INTEGER	atDrvrVersNum	Version number of AppleTalk, if installed		
INTEGER	sysVRefNum	Working directory reference number		

Source: Inside Macintosh, pages V-6 to V-8

See Also: 3.129. Toolbox Utility Routines

3.133. SysEnv Result Codes

Value	*Constant*	*Comments*
0	noErr	No error
-5500	envNotPresent	SysEnvirons trap is not present *
-5501	envBadVers	Nonpositive number was passed
-5502	envVersTooBig	Requested SysEnvirons call not available

Notes: * Must use System 4.1 or later.

Source: Inside Macintosh, page V-11

See Also: 3.129. Toolbox Utility Routines

3.134. Package Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
InitPack	Initializes a single package for use	I	PROCEDURE InitPack (packID:INTEGER);
InitAllPacks	Initializes all available packages	I	PROCEDURE InitAllPacks;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, page I-484

See Also:
3.008. InterfaceToolbox Routines
3.135. Supplied PACKs
3.136. Binary-Decimal Package Routines
3.137. International Package Routines
3.138. IntlORec Data Structure
3.139. Intl1Rec Data Structure
3.140. International Routine Selectors
3.141. International String Comparsion Result Codes
3.142. Standard File Package Routines
3.143. SFReply Data Structure
3.144. Standard File Routine Selectors
3.145. Standard File Predefined Constants
3.146. Standard File Resource IDs

3.135. Supplied PACKs

Value	*Constant*	*Comments*
0	listMgr	List manager
1	RESERVED	
2	dskInit	Disk initialization
3	stdFile	Standard file
4	flPoint	Floating-point arithmetic
5	trFunct	Transcendental functions
6	intUtil	International utilities
7	bdConv	Binary-decimal conversion
8	RESERVED	
9	RESERVED	
10	RESERVED	
11	RESERVED	
12	RESERVED	
13	RESERVED	
14	RESERVED	
15	RESERVED	

Source: Inside Macintosh, page IV-67

See Also: 3.134. Package Manager Routines
3.136. to 3.146. individual PACK information tables

3.136. Binary-Decimal Package Routines

Name	*Description*	*Defined **	*Pascal Definition*
NumToString	Converts number to string	I	PROCEDURE NumToString (theNum:LONGINT; VAR theString:Str255);
StringToNum	Converts string to number	I	PROCEDURE StringToNum (theString:Str255; VAR theNum:LONGINT);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-489 to I-490

See Also: 3.134. Package Manager Routines

3.137. International Package Routines

Name	*Description*	*Defined **	*Pascal Definition*
IUDateString	Returns date string from date and time values	I	PROCEDURE IUDateString (dateTime:LONGINT; form:DateForm; VAR result:Str255);
IUDatePString	Returns date string from date and time using resource	I	PROCEDURE IUDatePString (dateTime:LONGINT; form:DateForm; VAR result:Str255; intlParam:Handle);
IUTimeString	Returns time string from date and time values	I	PROCEDURE IUTimeString (dateTime:LONGINT; wantSeconds:BOOLEAN; VAR result:Str255);
IUTimePString	Returns time string from date and time using resource	I	PROCEDURE IUTimePString (dateTime:LONGINT; wantSeconds:BOOLEAN; VAR result:Str255; intlParam:Handle);
IUMetric	Returns TRUE if metric system being used	I	FUNCTION IUMetric : BOOLEAN;
IUGetIntl	Returns handle to intl resource	I	FUNCTION IUGetIntl (theID:INTEGER) : Handle;
IUSetIntl	Sets intl resource to data in intlParam	I	PROCEDURE IUSetIntl (refNum:INTEGER; theID:INTEGER; intlParam:Handle);
IUCompString	Compares strings using intl rules	I	FUNCTION IUCompString (aStr,bStr:Str255) : INTEGER;
IUMagString	Compares strings using intl rules	I	FUNCTION IUMagString (aPtr,bPtr:Ptr; aLen,bLen:INTEGER) : INTEGER;
IUEqualString	Compares strings for equality	I	FUNCTION IUEqualString (aStr,bStr:Str255) : INTEGER;
IUMagIDString	Compares strings for equality	I	FUNCTION IUMagIDString (aPtr,bPtr:Ptr; aLen,bLen:INTEGER) : INTEGER;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-504 to I-507

See Also: 3.134. Package Manager Routines
3.138. Intl0Rec Data Structure
3.139. Intl1Rec Data Structure
3.140. International Routine Selectors
3.141. International String Comparison Result Codes

3.138. IntlORec Data Structure

Size	*Name*	*Description*	*Domestic Values*
CHAR	decimalPt	Decimal point character	.
CHAR	thousSep	Thousands separator	,
CHAR	listSep	List separator	;
CHAR	currSym1	Currency symbol	$
CHAR	currSym2	Extra room for currency symbol	
CHAR	currSym3	Extra room for currency symbol	
Byte	currFmt	Currency format	Bit 7 = 1 leading integer zero Bit 6 = 1 trailing decimal zeroes Bit 5 = 1 minus sign for negative Bit 4 = 1 currency symbol leads
Byte	dateOrder	Order of date elements	Mdy = 0 Dmy = 1 Ymd = 2
Byte	shrtDateFmt	Format of short date string	Bit 7 = 1 century included Bit 6 = 1 leading zero for month Bit 5 = 1 leading zero for day
CHAR	dateSep	Date separator	/
Byte	timeCycle	Clock type used	0 = 24-hour, 1 = 12-hour
Byte	timeFmt	Format of time string	Bit 7 = 1 leading zero for hours Bit 6 = 1 leading zero for minutes Bit 5 = 1 leading zero for seconds
4 CHARs	mornStr	String for AM	'AM'
4 CHARs	eveStr	String for PM	'PM'
CHAR	timeSep	Time separator	:
CHAR	time1Stuff	Trailing string for 24-hour clock	
CHAR	time2Stuff	Trailing string for 24-hour clock	
CHAR	time3Stuff	Trailing string for 24-hour clock	
CHAR	time4Stuff	Trailing string for 24-hour clock	
CHAR	time5Stuff	Trailing string for 24-hour clock	
CHAR	time6Stuff	Trailing string for 24-hour clock	
CHAR	time7Stuff	Trailing string for 24-hour clock	
CHAR	time8Stuff	Trailing string for 24-hour clock	
Byte	metricSys	Metric system flag	0 = no, 255 = yes, metric
INTEGER	intl0Vers	International version	*

Notes: * One of the following constant values:

0	verUS
1	verFrance
2	verBritain
3	verGermany
4	verItaly
5	verNetherlands
6	verBelgiumLux
7	verSweden
8	verSpain
9	verDenmark
10	verPortugal
11	verFrCanada
12	verNorway
13	verIsrael
14	verJapan
15	verAustralia
16	verArabia
17	verFinland
18	verFrSwiss
19	verGrSwiss
20	verGreece
21	verIceland
22	verMalta
23	verCyprus
24	verTurkey
25	verYugoslavia

Source: Inside Macintosh, pages I-496 to I-499

See Also: 3.137. International Package Routines
3.139. Intl1Rec Data Structure

3.139. IntlIRec Data Structure

Size	*Name*	*Description*	*Values*
105 bytes	days	Names of days	Arranged as 7 arrays of 15 bytes each
180 bytes	months	Names of months	Arranged as 12 arrays of 15 bytes each
Byte	suppressDay	Day name appearance	0 = day name appears, 255 = doesn't
Byte	lngDateFmt	Long date format	*
Byte	dayLeading0	Leading 0 in day numbers	255 = leading 0, 0 = no leading 0
Byte	abbrLen	Length for abbreviating names	Length of abbreviated names
4 bytes	st0	Strings for long date format	
4 bytes	st1	Strings for long date format	
4 bytes	st2	Strings for long date format	
4 bytes	st3	Strings for long date format	
4 bytes	st4	Strings for long date format	
INTEGER	intl1Vers	Version information	See 3.138.
INTEGER	localRtn	Routine for localizing string comparison	See source

Notes: * 0 = st0 dayname st1 day st2 month st3 year st4.
255 = st0 dayname st1 month st2 day st3 year st4.

Source: Inside Macintosh, pages I-500 to I-501

See Also: 3.137. International Package Routines
3.138. IntlORec Data Structure

3.140. International Routine Selectors

Value	*Constant*
0	iuDateString
2	iuTimeString
4	iuMetric
6	iuGetIntl
8	iuSetIntl
10	iuMagString
12	iuMagIDString
14	iuDatePString
16	iuTimePString

Notes: Used in assembly language only.

Source: Inside Macintosh, page I-504

See Also: 3.137. International Package Routines

3.141. International String Comparison Result Codes

Value	*Meaning*
-1	aStr is less than bStr
0	aStr equals bStr
1	aStr is greater than bStr

Source: Inside Macintosh, page I-506

See Also: 3.137. International Package Routines

3.142. Standard File Package Routines

Name	*Description*	*Defined **	*Pascal Definition*
SFPutFile	Displays standard save file dialog box	I	PROCEDURE SFPutFile (where:Point; prompt:Str255; origName:Str255; dlgHook:ProcPtr; VAR reply:SFReply);
SFPPutFile	Displays save file dialog box, with user-defined template	I	PROCEDURE SFPPutFile (where:Point; pompt:Str255; origName:Str255; dlgHook:ProcPtr; VAR reply:SFReply; dlgID:INTEGER; filterProc:ProcPtr);
SFGetFile	Displays standard get file dialog box	I	PROCEDURE SFGetFile (where:Point; prompt:Str255; fileFilter:ProcPtr; numTypes:INTEGER; typeList:SFTypeList; dlgHook:ProcPtr; VAR reply:SFReply);
SFPGetFile	Displays get file dialog box, with user-defined template	I	PROCEDURE SFGetFile (where:Point; prompt:Str255; fileFilter:ProcPtr; numTypes:INTEGER; typeList:SFTypeList; dlgHook:ProcPtr; VAR reply:SFReply; dlgID:INTEGER; filterProc:ProcPtr);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages I-519 to I-526

See Also: 3.134. Package Manager Routines
3.143. SFReply Data Structure
3.144. Standard File Routine Selectors
3.146. Standard File Resource IDs

3.143. SFReply Data Structure

Size	*Name*	*Comments*
BOOLEAN	good	FALSE = ignore command
BOOLEAN	copy	NOT USED
OSType	fType	File type
INTEGER	vRefNum	Volume reference number
INTEGER	version	Version number of file
63 bytes	fName	Name of file

Source: Inside Macintosh, page I-519

See Also: 3.142. Standard File Package Routines

3.144. Standard File Routine Selectors

Value	*Constant*
1	sfPutFile
2	sfGetFile
3	sfPPutFile
4	sfPGetFile

Notes: Normally used only from assembly language.

Source: Inside Macintosh, page I-519

See Also: 3.142. Standard File Package Routines

3.145. Standard File Predefined Constants

For PutFile

Value	*Constant*	*Comments*
1	putSave	Save button
2	putCancel	Cancel button
5	putEject	Eject button
6	putDrive	Drive button
7	putName	EditText item (filename)

For GetFile

1	getOpen	Open button
3	getCancel	Cancel button
5	getEject	Eject button
6	getDrive	Drive button
7	getNmList	UserItem (filename list)
8	getScroll	UserItem (scroll bar)

Source: Inside Macintosh, pages I-522 and I-526

See Also: 3.142. Standard File Package Routines

3.146. Standard File Resource IDs

Value	*Description*
-3994	Alert for Disk Not Found
-3995	Alert for System Error
-3996	Alert for Existing File
-3997	Alert for Locked Disk

Source: Inside Macintosh, pages I-522 to I-523

See Also: 3.142. Standard File Package Routines

3.147. Palette Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
ActivatePalette	Provides new color environment when window changes	V	PROCEDURE ActivatePalette (srcWindow:WindowPtr);
AnimateEntry	Changes RGB value of entry in palette	V	PROCEDURE AnimateEntry (dstWindow:WindowPtr; dstEntry:INTEGER; srcRGB:RGBColor);
AnimatePalette	Changes range of entries in palette	V	PROCEDURE AnimatePalette (dstWindow:WindowPtr; srcCTab:CTabHandle; srcIndex,dstEntry,dstLength:INTEGER);
CTab2Palette	Copies fields from color table to palette	V	PROCEDURE CTab2Palette (srcCTab:CTabHandle; dstPalette:PaletteHandle; srcUsage,srcTolerance:INTEGER);
DisposePalette	Releases palette object from memory	V	PROCEDURE DisposePalette (srcPalette:PaletteHandle);
GetEntryColor	Returns color of a palette entry	V	PROCEDURE GetEntryColor (srcPalette:PaletteHandle; srcEntry:INTEGER; VAR dstRGB:RGBColor);
GetEntryUsage	Returns usage fields of palette entry	V	PROCEDURE GetEntryUsage (srcPalette:PaletteHandle; srcEntry:INTEGER; VAR dstUsage,dstTolerance:INTEGER);
GetNewPalette	Returns handle to palette object from resource file	V	FUNCTION GetNewPalette (paletteID:INTEGER) : PaletteHandle;
GetPalette	Returns handle to palette associated with window	V	FUNCTION GetPalette (srcWindow:WindowPtr) : PaletteHandle;
InitPalettes	Initializes Palette Manager	V	PROCEDURE InitPalettes;
NewPalette	Returns handle to new palette object	V	FUNCTION NewPalette (entries:INTEGER; srcColors:CTabHandle; srcUsage, srcTolerance:INTEGER) : PaletteHandle;
Palette2CTab	Copies colors from palette to color table	V	PROCEDURE Palette2CTab (srcPalette:PaletteHandle; dstCTab:CTabHandle);
PmBackColor	Sets RGB/index backcolor fields of cGrafPort	V	PROCEDURE PmBackColor (dstEntry:INTEGER);
PmForeColor	Sets RGB/index forecolor fields of cGrafPort	V	PROCEDURE PmForeColor (dstEntry:INTEGER);
SetEntryColor	Modifies color of a palette entry	V	PROCEDURE SetEntryColor (dstPalette:PaletteHandle; dstEntry:INTEGER; srcRGB:RGBColor);
SetEntryUsage	Modifies usage fields of palette entry	V	PROCEDURE SetEntryUsage (dstPalette:PaletteHandle; dstEntry:INTEGER; srcUsage,srcTolerance:INTEGER);
SetPalette	Changes palette associated with window	V	PROCEDURE SetPalette (dstWindow:WindowPtr; srcPalette:PaletteHandle; cUpdates:BOOLEAN);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-161 to V-167

See Also:
3.008. Interface Toolbox Managers
3.034. Color QuickDraw Routines
3.035. RGBColor Data Structure
3.038. ColorGrafPort Data Structure
3.148. Palette Manager Color Methods
3.149. ColorInfo Data Structure
3.150. Palette Data Structure

3.148. Palette Manager Color Methods

Value	*Constant*	*Comments*
$0000	pmCourteous	No special properties, color QuickDraw picks pixel values
$0001	pmDitherered	RESERVED (not currently implemented)
$0002	pmTolerant	Changes color enrivonment unless color is within specified tolerance value
$0004	pmAnimated	USED INTERNALLY (used by palette only)
$0008	pmExplicit	Always generates corresponding entry in device's color table

Source: Inside Macintosh, pages V-154 to V-157

See Also: 3.147. Palette Manager Routines

3.149. ColorInfo Data Structure

Size	*Name*	*Comments*
RGBColor	ciRGB	Absolute RGB values
INTEGER	ciUsage	See 3.148.
INTEGER	ciTolerance	Tolerance value
INTEGER	ciFlags	USED INTERNALLY
LONGINT	ciPrivate	USED INTERNALLY

Source: Inside Macintosh, page V-159

See Also: 3.035. RGBColor Data Structure
3.147. Palette Manager Routines
3.148. Palette Manager Color Methods

3.150. Palette Data Structure

Size	*Name*	*Values/Comments*
INTEGER	pmEntries	Number of entries in pmInfo
INTEGER array	pmDataFields	USED INTERNALLY by Palette Manager (7 INTEGERS)
ColorInfo array	pmInfo	ColorInfo entries for palette

Source: Inside Macintosh, page V-159

See Also: 3.147. Palette Manager Routines
3.149. ColorInfo Data Structure

3.151. List Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
LActivate	Sets status of list	IV	PROCEDURE LActivate (act:BOOLEAN; lHandle:ListHandle);
LAddColumn	Inserts column(s) into list	IV	FUNCTION LAddColumn (count,colNum:INTEGER; lHandle:ListHandle) : INTEGER;
LAddRow	Inserts row(s) into list	IV	FUNCTION LAddRow (count,rowNum:INTEGER; lHandle:ListHandle) : INTEGER;
LAddToCell	Appends data to cell	IV	PROCEDURE LAddToCell (dataPtr:Ptr; dataLen:INTEGER; theCell:Cell; lHandle:ListHandle);
LAutoScroll	Scrolls list to selected cell	IV	PROCEDURE LAutoScroll (lHandle:ListHandle);
LCellSize	Sets cell size, updates visible rectangle	IV	PROCEDURE LCellSize (cSize:Point; lHandle:ListHandle);
LClick	Returns TRUE if doubleclick in cell	IV	FUNCTION LClick (pt:Point; modifiers:INTEGER; lHandle:ListHandle) : BOOLEAN;
LClrCell	Clears contents of cell	IV	PROCEDURE LClrCell (theCell:Cell; lHandle:ListHandle);
LDelColumn	Deletes column(s) from list	IV	PROCEDURE LDelColumn (count,colNum:INTEGER; lHandle:ListHandle);
LDelRow	Deletes row(s) from list	IV	PROCEDURE LDelRow (count,rowNum:INTEGER; lHandle:ListHandle);
LDispose	Releases list from memory	IV	PROCEDURE LDispose (lHandle:ListHandle);
LDoDraw	Sets LM drawing mode state	IV	PROCEDURE LDoDraw (drawIt:BOOLEAN; lHandle:ListHandle);
LDraw	Draws cell after entry or selection	IV	PROCEDURE LDraw (theCell:Cell; lHandle:ListHandle);
LFind	Returns offset/length of cell's data	IV	PROCEDURE LFind (VAR offset,len:INTEGER; theCell:Cell; lHandle:ListHandle);
LGetCell	Returns copy of cells contents	IV	PROCEDURE LGetCell (dataPtr:Ptr; VAR dataLen:INTEGER; theCell:Cell; lHandle:ListHandle);
LGetSelect	Returns TRUE if cell selected	IV	FUNCTION LGetSelect (next:BOOLEAN; VAR theCell:Cell; lHandle:ListHandle) : BOOLEAN;
LLastClick	Returns cell coordinates of last click	IV	FUNCTION LLastClick (lHandle:ListHandle) : Cell;
LNew	Creates a new list	IV	FUNCTION LNew (rView,dataBounds:Rect; cSize:Point; theProc:INTEGER; theWindow:WindowPtr; drawIt,hasGrow,scrollHoriz,scrollVert:BOOLEAN) : ListHandle;
LNextCell	Returns next cell in list	IV	FUNCTION LNextCell (hNext,vNext:BOOLEAN; VAR theCell:Cell; lHandle:ListHandle) : BOOLEAN;
LRect	Returns local coordinates of cell	IV	PROCEDURE LRect (VAR cellRect:Rect; theCell:Cell; lHandle:ListHandle);
LScroll	Scrolls list	IV	PROCEDURE LScroll (dCols,dRows:INTEGER; lHandle:ListHandle);
LSearch	Searchs for cell with data	IV	FUNCTION LSearch (dataPtr:Ptr; dataLen:INTEGER; searchProc:Ptr; VAR theCell:Cell; lHandle:ListHandle) : BOOLEAN;
LSetCell	Modifies cell contents	IV	PROCEDURE LSetCell (dataPtr:Ptr; dataLen:INTEGER; theCell:Cell; lHandle:ListHandle);
LSetSelect	Selects cell and redraws it	IV	PROCEDURE LSetSelect (setIt:BOOLEAN; theCell:Cell; lHandle:ListHandle);
LSize	Resizes list to changed window size	IV	PROCEDURE LSize (listWidth,listHeight:INTEGER; lHandle:ListHandle);
LUpdate	Redraws visible cells/controls	IV	PROCEDURE LUpdate (theRgn:RgnHandle; lHandle:ListHandle);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages IV-269 to IV-277

See Also: 3.008. Interface Toolbox Routines
3.152. ListRec Data Structure
3.153. List Selection Flag Byte
3.154. List Manager Masks
3.155. List Manager Messages

3.152. ListRec Data Structure

Size	*Name*	*Comments*
Rect	rView	Display rectangle for list
GrafPtr	port	GrafPort for list
Point	indent	Indent distance, in pixels
Point	cellSize	Cell size, in pixels
Rect	visible	Visible cell boundary
ControlHandle	vScroll	Vertical scroll bar, if used
ControlHandle	hScroll	Horizontal scroll bar, if used
SignedByte	selFlags	Selection flags (see 3.153.)
BOOLEAN	lActive	TRUE if list is active
SignedByte	lReserved	RESERVED
SignedByte	listFlags	Automatic scrolling flags
LONGINT	clikTime	Time of last click
Point	clikLoc	Position of last click
Point	mouseLoc	Current mouse location
Ptr	lClikLoop	Routine for LClick
Cell	lastClick	Last cell clicked
LONGINT	refCon	Reference value for list
Handle	listDefProc	Definition procedure for list
Handle	userHandle	User-defined
Rect	dataBounds	Boundary of allocated cells
DataHandle	cells	Cell data
INTEGER	maxIndex	USED INTERNALLY
INTEGER array	cellArray	Offsets to cell data

Source: Inside Macintosh, pages IV-262 to IV-266

See Also: 3.151. List Manager Routines
3.153. List Selection Flag Byte

3.153. List Selection Flag Byte

Bit Numbers

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Name*	*Values*
X								lOnlyOne	1 = one selection, 0 = multiple selections
	X							lExtendDrag	1 = drag without shift key, 0 = drag only with shift key
		X						lNoDisjoint	1 = turn off multiple selections with click
			X					lNoExtend	1 = don't extend shift selections, 0 = extend shift selections
				X				lNoRect	1 = don't grow selections as rectangles, 0 = grow them as rectangles
					X			lUseSense	1 = shift use sense of first cell
						X		lNoNilHilite	1 = don't highlight empty cells, 0 = highlight empty cells
							X	RESERVED	

Source: Inside Macintosh, pages IV-266 to IV-268

See Also: 3.151. List Manager Routines
3.152. ListRec Data Structure

3.154. List Manager Masks

Value	*Constant*	*Comments*
2	lDoVAutoscroll	Allow automatic scrolling in vertical direction
1	lDoHAutoscroll	Allow automatic scrolling in horizontal direction
-128	lOnlyOne	See 3.153.
64	lExtendDrag	See 3.153.
32	lNoDisjoint	See 3.153.
16	lNoExtend	See 3.153.
8	lNoRect	See 3.153.
4	lUseSense	See 3.153.
2	lNoNilHilite	See 3.153.

Source: Inside Macintosh, page IV-278

See Also: 3.151. List Manager Routines
3.152. ListRec Data Structure
3.153. List Selection Flag Byte

3.155. List Manager Messages

Value	*Constant*	*Comments*
0	lInitMsg	Initialize list, allocate space, set default values
1	lDrawMsg	Draw cell data
2	lHiliteMsg	Select or deselect a cell
3	lCloseMsg	Dispose of cell list

Source: Inside Macintosh, page IV-278

See Also: 3.151. List Manager Routines

3.156. Operating System Calls

Manager	*ROM Introduced **	*See Tables*
Memory Manager	64K	3.157. to 3.167.
Segment Loader	64K	3.168. to 3.171.
Event Manager	64K	3.172. to 3.180.
File Manager	64K	3.181. to 3.199.
Device Manager	64K	3.200. to 3.228.
Disk Driver	64K	3.229. to 3.233.
SCSI Manager	64K	3.234. to 3.239.
Sound Manager	64K	3.240. to 3.244.
Serial Drivers	64K	3.245. to 3.251.
System Error Handler	64K	3.252. to 3.253.
Operating System Utilities	64K	3.254. to 3.257.
Printing Manager	64K	3.258. to 3.270.
Slot Manager	256K (Mac II)	3.271. to 3.280.
Start Manager	256K	3.281. to 3.284.
Shutdown Manager	256K	3.285. to 3.286.
ADB Manager	256K	3.287. to 3.291.
Video Driver	256K (Mac II)	3.292. to 3.294.
Color Picker	256K (Mac II)	3.295. to 3.298.
Vertical Retrace Manager	64K	3.299. to 3.301.
AppleTalk Manager	64K	3.302. to 3.322.
Disk Initialization Package	64K	3.323. to 3.325.
Time Manager	64K	3.326. to 3.327.
Deferred Task Manager	256K	3.328. to 3.329.
SANE	64K	3.330. to 3.337.

Notes: * ROM where manager first appeared.

Source: Technical Introduction to the Macintosh Family, pages 16 to 17, 21 to 23

See Also: 3.008. Interface Toolbox Routines
Tables listed above

3.157. Memory Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
ApplicZone	Returns pointer to original application heap zone	II	FUNCTION ApplicZone : THz;
BlockMove	Moves block of bytes	II	PROCEDURE BlockMove (sourcePtr,destPtr:Ptr; byteCount:Size);
CompactMem	Compacts heap zone	II	FUNCTION CompactMem (cbNeeded: Size) : Size;
DisposHandle	Releases memory used by handle	II	PROCEDURE DisposHandle (h:Handle);
DisposPtr	Releases memory used by pointer	II	PROCEDURE DisposPtr (p:Ptr);
EmptyHandle	Purges relocatable block sets pointer to NIL	II	PROCEDURE EmptyHandle (h:Handle);
FreeMem	Returns amount of free space in heap, in bytes	II	FUNCTION FreeMem : LONGINT;
GetApplLimit	Returns application heap limit	II	FUNCTION GetApplLimit : Ptr;
GetHandleSize	Returns logical size of block, in bytes	II	FUNCTION GetHandleSize (h:Handle) : Size;
GetPtrSize	Returns logical size of nonrelocatable block	II	FUNCTION GetPtrSize (p:Ptr) : Size;
GetZone	Returns pointer to current heap zone	II	FUNCTION GetZone : THz;
GZSaveHnd	Returns handle to relocatable block	II	FUNCTION GZSaveHnd : Handle;
HandleZone	Returns pointer to heap zone with block	II	FUNCTION HandleZone (h:Handle) : THz;
HClrRBit	Clears resource flag of relocatable block's master pointer	IV	PROCEDURE HClrBit (h:Handle);
HGetState	Returns byte of master pointer flags	IV	FUNCTION HGetState (h:Handle) : SignedByte;
HLock	Locks relocatable block	II	PROCEDURE HLock (h:Handle);
HNoPurge	Marks relocatable block as unpurgeable	II	PROCEDURE HNoPurge (h:Handle);
HPurge	Marks relocatable block as purgeable	II	PROCEDURE HPurge (h:Handle);
HSetRBit	Sets resource flag of relocatable block's master pointer	IV	PROCEDURE HSetRBit (h:Handle);
HSetState	Modifies byte of master pointer flags	IV	PROCEDURE HSetState (h:Handle; flags:SignedByte);
HUnlock	Unlocks relocatable block	II	PROCEDURE HUnlock (h:Handle);
InitApplZone	Initializes application heap zone	II	PROCEDURE InitApplZone;
InitZone	Creates new heap zone	II	PROCEDURE InitZone (pGrowZone:ProcPtr; cMoreMasters:INTEGER; limitPtr,startPtr:Ptr);
MaxApplZone	Expands zone to application limit	II	PROCEDURE MaxApplZone;
MaxBlock	Returns maximum contiguous space after compact, in bytes	IV	FUNCTION MaxBlock : LONGINT;
MaxMem	Purges blocks from zone and compacts	II	FUNCTION MaxMem (VAR grow: Size) : Size;
MemError	Returns OSErr for last Memory Manager routine	II	FUNCTION MemError : OSErr;
MoreMasters	Allocates additional block of master pointers	II	PROCEDURE MoreMasters;
MoveHHi	Moves relocatable block to top of current heap	II	PROCEDURE MoveHHi (h:Handle);
NewEmptyHandle	Returns NIL master pointer	IV	FUNCTION NewEmptyHandle : Handle;
NewHandle	Allocates new relocatable block in current heap	II	FUNCTION NewHandle (logicalSize:Size) : Handle;
NewPtr	Allocates nonrelocatable block	II	FUNCTION NewPtr (logicalSize:Size) : Ptr;
PtrZone	Returns pointer to heap zone with nonrelocatable block	II	FUNCTION PtrZone (p:Ptr) : THz;
PurgeMem	Purges cbNeeded blocks from current heap	II	PROCEDURE PurgeMem (cbNeeded:Size);
PurgeSpace	Returns total possible space that could be obtained from purge, in bytes	IV	PROCEDURE PurgeSpace (VAR total,contig:LONGINT);
ReallocHandle	Allocates new relocatable block	II	PROCEDURE ReallocHandle (h:Handle; logicalSize:Size);
RecoverHandle	Returns handle to relocatable block	II	FUNCTION RecoverHandle (p:Ptr) : Handle;
ResrvMem	Creates free space for block cbNeeded	II	PROCEDURE ResrvMem (cbNeeded:Size);
SetApplBase	Changes starting address of application heap zone	II	PROCEDURE SetApplBase (startPtr:Ptr);
SetApplLimit	Sets application heap limit	II	PROCEDURE SetApplLimit (zoneLimit:Ptr);
SetGrowZone	Sets current heap zone's grow zone	II	PROCEDURE SetGrowZone (growZone:ProcPtr);
SetHandleSize	Modifies logical size of block	II	PROCEDURE SetHandleSize (h:Handle; newSize:Size);
SetPtrSize	Modifies logical size of nonrelocatable block	II	PROCEDURE SetPtrSize (p:Ptr; newSize:Size);
SetZone	Sets current heap zone	II	PROCEDURE SetZone (hz: THz);
StackSpace	Returns amount of stack space remaining	IV	FUNCTION StackSpace : LONGINT;
SystemZone	Returns pointer to system heap zone	II	FUNCTION SystemZone : THz;
TopMem	Returns pointer to end of RAM on 128/512/XL	II	FUNCTION TopMem : Ptr;

(continued)

Table 3.157. Continued

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-27 to II-44, IV-77 to IV-80

See Also: 3.156. Operating System Calls
3.158. Memory Organization for 128/512/Plus
3.159. Memory Organization for XL
3.160. Memory Organization for SE
3.161. Memory Organization for II
3.162. Zone Data Structure
3.163. Zone Block Header Data Structure
3.164. Tab Byte Format
3.165. Master Pointer Structure
3.166. InitZone Parameter Block
3.167. Memory Manager Result Codes

3.158. Memory Organization for 128/512/Plus

Global Variables	*Memory Usage*	*Size, in Bytes*
MemTop	NOT USED	28
	Sound buffer (main)	740
	SystemError Handler	128
ScrnBase	Screen buffer (main)	21888
	NOT USED	796
	Sound buffer (alternate)	740
	NOT USED	9344
BufPtr	Screen buffer (alternate)	21888
	Jump table	Varies
CurrentA5	Application parameters	32
	Application globals	Varies
CurStackBase	QuickDraw globals	206
	Stack	Varies
ApplZone	Application heap	
SysZone	System heap	Varies
	System globals	2816

Notes: Application space extends from ApplZone to BufPtr.

Source: Inside Macintosh, page II-19

See Also: 3.159. Memory Organization for XL
3.160. Memory Organization for SE
3.161. Memory Organization for II

3.159. Memory Organization for XL

Global Variables	*Memory Usage*	*Size, in Bytes*
ScrnBase	Screen buffer	32768
MemTop/BufPtr	Hardware interface	
	Jump table	Varies
CurrentA5	Application parameters	32
	Application globals	Varies
CurStackBase	QuickDraw globals	206
	Stack	
		Varies
ApplZone	Application heap	
SysZone	System heap	Varies
	System globals	2816

Notes: Application space extends from ApplZone to BufPtr.

Source: Inside Macintosh, page II-21

See Also: 3.158. Memory Organization for 128/512/Plus
3.160. Memory Organization for SE
3.161. Memory Organization for II

3.160. Memory Organization for SE

Global Variables	*Memory Usage*	*Size, in Bytes*
MemTop	NOT USED	28
	Sound buffer (main)	740
	SystemError Handler	128
ScrnBase	Screen buffer (main)	21888
	NOT USED	
BufPtr	Screen buffer (alternate)	21888
	Jump table	Varies
CurrentA5	Application parameters	32
	Application globals	Varies
CurStackBase	QuickDraw globals	206
	Stack	
		Varies
ApplZone	Application heap	
SysZone	System heap	Varies
	System globals	2816
	Exception vectors	256

Notes: Application space extends from ApplZone to BufPtr.

Source: Technical Introduction to the Macintosh Family, pages 140 to 141

See Also: 3.158. Memory Organization for 128/512/Plus
3.159. Memory Organization for XL
3.161. Memory Organization for II

3.161. Memory Organization for II

Global Variables	*Memory Usage*	*Size, in Bytes*
MemTop	NOT USED	128
	System error handler	
BufPtr	NOT USED	
	Jump table	Varies
CurrentA5	Application parameters	32
	Application globals	Varies
CurStackBase	QuickDraw globals	206
	Stack	Varies
ApplZone	Application heap	
SysZone	System heap	Varies
	System globals	2816

Notes: Application space extends from ApplZone to BufPtr.

Source: Technical Introduction to the Macintosh Family, pages 142 to 143

See Also: 3.158. Memory Organization for 128/512/Plus
3.159. Memory Organization for XL
3.160. Memory Organization for SE

3.162. Zone Data Structure

Size	*Name*	*Values/Comments*
Ptr	bkLim	Zone trailer block
Ptr	purgePtr	USED INTERNALLY
Ptr	hFstFree	First free master pointer
LONGINT	zcbFree	Number of free bytes
ProcPtr	gzProc	Grow zone function
INTEGER	moreMast	Master pointers to allocate
INTEGER	flags	USED INTERNALLY
INTEGER	cntRel	NOT USED
INTEGER	maxRel	NOT USED
INTEGER	cntNRel	NOT USED
INTEGER	maxNRel	NOT USED
INTEGER	cntEmpty	NOT USED
INTEGER	cntHandles	NOT USED
LONGINT	minCBFree	NOT USED
ProcPtr	purgeProc	Purge warning procedure
Ptr	sparePtr	USED INTERNALLY
Ptr	allocPtr	USED INTERNALLY
INTEGER	heapData	First usable byte in zone

Source: Inside Macintosh, pages II-22 to II-23

See Also: 3.157. Memory Manager Routines

3.163. Zone Block Header Data Structure

Size	*Name*	*Comments*
1 byte	tagByte	See 3.164.
3 bytes	blockSize	*

Notes: * For relocatable block: relative handle; for nonrelocatable block: pointer to heap zone.

Source: Inside Macintosh, page II-24

See Also: 3.157. Memory Manager Routines
3.164. Tag Byte Format

3.164. Tag Byte Format

Bit Numbers

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Values*
X	X							00 = free block, 01 = nonrelocatable, 10 = relocatable
		X	X					NOT USED
				X	X	X	X	Size correction (unused bytes at end of block)

Source: Inside Macintosh, page II-24

See Also: 3.163. Zone Block Header Data Structure

3.165. Master Pointer Structure

Bit Numbers

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Values*
X								Lock bit (1 = locked)
	X							Purge bit (1 = purgeable)
		X						Resource bit (1 = contains resource info)
			X	X	X	X	X	NOT USED

Notes: Flag byte listed above is followed by address of block's contents.

Source: Inside Macintosh, page II-25

See Also: 3.157. Memory Manager Routines

3.166. InitZone Parameter Block

Size	*Name*	*Comments*
Ptr	startPtr	First byte of new zone
Ptr	limitPtr	First byte beyond end of zone
word	cMoreMasters	How many master pointers to allocate
Ptr	pGrowZone	Pointer to grow zone function

Notes: Used to call _InitZone from assembly language.

Source: Inside Macintosh, page II-29

See Also: 3.157. Memory Manager Routines

3.167. Memory Manager Result Codes

Value	*Constant*	*Comments*
-117	memLockedErr	Block is locked
-112	memPurErr	Block is locked, can't purge
-111	memWZErr	Block is free
-109	nilHandleErr	Block has NIL master pointer
-108	memFullErr	Out of room in heap zone
-99	memROZErr	Block is read-only
0	noErr	No error

Source: Inside Macintosh, pages II-46, IV-81

See Also: 3.157. Memory Manager Routines

3.168. Segment Loader Routines

Name	*Description*	*Defined **	*Pascal Definition*
ClrAppFiles	Changes Finder information about file	II	PROCEDURE ClrAppFiles (index:INTEGER);
CountAppFiles	Returns information about documents started with application	II	PROCEDURE CountAppFiles (VAR message:INTEGER; VAR count:INTEGER);
ExitToShell	Exits application to Finder	II	PROCEDURE ExitToShell;
GetAppFiles	Returns information about single document	II	PROCEDURE GetAppFiles (index:INTEGER; VAR theFile:AppFile);
GetAppParms	Returns information about current application	II	PROCEDURE GetAppParms (VAR apName:Str255; VAR apRefNum:INTEGER; VAR apParam:Handle);
UnloadSeg	Unloads segment (makes purgeable/relocatable)	II	PROCEDURE UnloadSeg (routineAddr:Ptr);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-57 to II-59

See Also: 3.156. Operating System Calls
3.169. Finder Information Data Structure
3.170. AppFile Data Structure
3.171. Jump Table Formats

3.169. Finder Information Data Structure

Size	*Name*	*Comments*
Word	message	0 = open doc, 1 = print doc
Word	count	Number of documents (0 = none)
Varies	docInfo	Information about documents

--------->

For Each Document

Word	vRefNum	0 = default volume
Long word	fileType	Four-character file type
Byte	versNum	Version number
Byte	NOT USED	
Byte	nameLength	Length of filename, in bytes
Chars	name	File name string

Source: Inside Macintosh, pages II-55 to II-56

See Also: 3.168. Segment Loader Routines

3.170. AppFile Data Structure

Size	*Name*	*Comments*
INTEGER	vRefNum	Volume reference number (0 = default)
OSType	fType	Four-byte file type string
INTEGER	versNum	Version number
Str255	fName	File name

Source: Inside Macintosh, page II-58

See Also: 3.168. Segment Loader Routines
3.169. Finder Information Data Structure

3.171. Jump Table Formats

Unloaded State:

Size	*Name*	*Comments*
Word	segOffset	Offset to beginning of segment
Long word	segNumMove	Instruction to move segment number onto stack
Word	loadSegTrap	LoadSeg trap instruction

Loaded State:

Size	*Name*	*Comments*
Word	segNum	Segment number
6 bytes	segInstJump	Jump instruction to address of segment

Notes: See note in source about structure of segment header.

Source: Inside Macintosh, pages II-60 to II-62

See Also: 3.168. Segment Loader Routines

3.172. Event Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
FlushEvents	Removes events from queue	II	PROCEDURE FlushEvents (eventMask,stopMask:INTEGER);
GetEvQHdr	Returns pointer to head of event queue	II	FUNCTION GetEvQHdr : QHdrPtr;
GetOSEvent	Returns next available event of type(s)	II	FUNCTION GetOSEvent (eventMask:INTEGER; VAR theEvent:EventRecord) : BOOLEAN;
OSEventAvail	Returns next event, but doesn't remove it from queue	II	FUNCTION OSEventAvail (eventMask:INTEGER; VAR theEvent:EventRecord) : BOOLEAN;
PostEvent	Places new event in queue	II	FUNCTION PostEvent (eventCode:INTEGER; eventMsg:LONGINT) : OSErr;
PPostEvent	Posts application-defined event in queue	IV	FUNCTION PPostEvent (eventCode:INTEGER; eventMsg:LONGINT; VAR qE1:EvQE1) : OSErr;
SetEventMask	Sets system event mask	II	PROCEDURE SetEventMask (theMask:INTEGER);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-68 to II-71, IV-85

See Also:
3.156. Operating System Calls
3.173. EventRecord Data Structure
3.174. EvQE1 Data Structure
3.175. Event Codes
3.176. Keyboard Event Message
3.177. KeyTrans Data Structure
3.178. Event Masks
3.179. Event Modifier Flags
3.180. PostEvent Result Codes

3.173. EventRecord Data Structure

Size	*Name*	*Comments*
INTEGER	what	See 3.070.
LONGINT	message	*
LONGINT	when	Number of ticks since startup
Point	where	Mouse location
INTEGER	modifiers	See 3.072.

Notes: * Message field varies depending on what type of event occurred:

Event	*Message field contains*
Keyboard	Character code, key code in LO word, ADB address in HO word
Activate, update	Pointer to window
Disk-inserted	Drive number in LO word, File Manager result code in HO word
Mouse-down, mouse-up, null	UNDEFINED
Network	Handle to parameter block
Device driver	Varies depending upon driver
Application-defined	Varies depending upon application

Source: Inside Macintosh, pages I-249, II-73, V-193 to V-194

See Also: 3.067. Toolbox Event Manager Routines
3.172. Event Manager Routines
3.175. Event Codes
3.176. Keyboard Event Message
3.178. Event Masks
3.179. Event Modifier Flags

3.174. EvQE1 Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Next queue entry
INTEGER	qType	Type of queue
INTEGER	evtQWhat	Event code
LONGINT	evtQMessage	Event message
LONGINT	evtQWhen	Ticks since startup
Point	evtQWhere	Mouse location of event
INTEGER	evtQModifiers	Modifier flags

Source: Inside Macintosh, page II-71

See Also: 3.172. Event Manager Routines

3.175. Event Codes

Value	*Constants*
0	nullEvent
1	mouseDown
2	mouseUp
3	keyDown
4	keyUp
5	autoKey
6	updateEvt (used by toolbox only; see 3.067.)
7	diskEvt
8	activateEvt (used by toolbox only; see 3.067.)
9	NOT USED
10	networkEvt
11	driverEvt
12	app1Evt
13	app2Evt
14	app3Evt
15	app4Evt

Source: Inside Macintosh, page II-72

See Also: 3.067. Toolbox Event Manager Routines
3.172. Event Manager Routines
3.173. EventRecord Data Structure

3.176. Keyboard Event Message

Size	*Name*	*Comments*
Byte	RESERVED	
Byte	adbAddr	Address of ADB keyboard used
Byte	virtKeyCode	Virtual key code received
Byte	asciiChar	ASCII character received

Source: Inside Macintosh, page V-194

See Also: 3.172. Event Manager Routines
3.173. EventRecord Data Structure

3.177. KeyTrans Data Structure

Size	*Name*	*Values/Comments*
Byte	RESERVED	
Byte	ASCII 1	Primary ASCII character received
Byte	RESERVED	
Byte	ASCII 2	Secondary ASCII character *

Notes: * For example, may be special accent to add to character.

Source: Inside Macintosh, page V-195

See Also: 3.172. Event Manager Routines
3.173. EventRecord Data Structure

3.178. Event Masks

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
X																App4Mask
	X															App3Mask
		X														App2Mask
			X													App1Mask
				X												Device driver
					X											Network
						X										NOT USED
							X									Activate
								X								Disk-inserted
									X							Update
										X						Auto-key
											X					Key-up
												X				Key-down
													X			Mouse-up
														X		Mouse-down
															X	NOT USED

Notes: To get every event, use a mask of -1 ($FFFF).

Source: Inside Macintosh, page II-72

See Also: 3.172. Event Manager Routines

3.179. Event Modifier Flags

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description	Values
X	X	X														RESERVED	
			X													Control key	1 = down, 0 = up
				X												Option key	1 = down, 0 = up
					X											Caps Lock key	1 = down, 0 = up
						X										Shift key	1 = down, 0 = up
							X									Command key	1 = down, 0 = up
								X								Mouse button	1 = up, 0 = down
									X	X	X	X	X	X		RESERVED	
															X	Window activation	1 = activated, 0 = deactivated

Source: Inside Macintosh, pages I-252 to I-253, II-73, V-194

See Also: 3.172. Event Manager Routines
3.173. EventRecord Data Structure

3.180. PostEvent Result Codes

Value	Constant	Comments
0	noErr	No error
1	evtNotEnb	Event type not in system event mask

Source: Inside Macintosh, page II-73

See Also: 3.172. Event Manager Routines

3.181. File Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
Allocate	Adds new empty bytes to end of file	II, IV	FUNCTION Allocate (refNum:INTEGER; VAR count:LONGINT) : OSErr;
Create	Creates new file	II, IV	FUNCTION Create (fileName:Str255; vRefNum:INTEGER; creator:OSType; fileType:OSType) : OSErr;
Eject	Flushes volume, ejects it	II, IV	FUNCTION Eject (volName: StringPtr; vRefNum:INTEGER) : OSErr;
FlushVol	Writes contents of volume buffer and vol info	II, IV	FUNCTION FlushVol (volName:StringPtr; vRefNum:INTEGER) : OSErr;
FSClose	Closes file access path	II, IV	FUNCTION FSClose (refNum:INTEGER) : OSErr;
FSDelete	Removes file from volume	II, IV	FUNCTION FSDelete (fileName:Str255; vRefNum:INTEGER) : OSErr;
FSOpen	Opens data fork file access path	II, IV	FUNCTION FSOpen (fileName:Str255; vRefNum:INTEGER; VAR refNum:INTEGER) : OSErr;
FSRead	Reads bytes from file	II, IV	FUNCTION FSRead (refNum:INTEGER; VAR count:LONGINT; buffPtr:Ptr) : OSErr;
FSWrite	Writes bytes to file	II, IV	FUNCTION FSWrite (refNum:INTEGER; VAR count:LONGINT; buffPtr:Ptr) : OSErr;
GetEOF	Returns logical end-of-file position	II, IV	FUNCTION GetEOF (refNum:INTEGER; VAR logEOF:LONGINT) : OSErr;
GetFInfo	Returns Finder information for file	II, IV	FUNCTION GetFInfo (fileName:Str255; vRefNum:INTEGER; VAR fndrInfo:FInfo) : OSErr;
GetFPos	Returns filePos in open file	II, IV	FUNCTION GetFPos (refNum:INTEGER; VAR filePos:LONGINT) : OSErr;
GetVInfo	Returns information about volume	II, IV	FUNCTION GetVInfo (drvNum:INTEGER; volName:StringPtr; VAR vRefNum:INTEGER; VAR freeBytes:LONGINT) : OSErr;
GetVol	Returns name of default volume and its refNum	II, IV	FUNCTION GetVol (volName:StringPtr; VAR vRefNum:INTEGER) : OSErr;
GetVRefNum	Returns volume reference number	II, IV	FUNCTION GetVRefNum (pathRefNum:INTEGER; VAR vRefNum:INTEGER) : OSErr;
OpenRF	Opens resource fork file access path	II, IV	FUNCTION OpenRF (fileName:Str255; vRefNum:INTEGER; VAR refNum:INTEGER) : OSErr;
Rename	Changes file's name	II, IV	FUNCTION Rename (oldName:Str255; vRefNum:INTEGER; newName:Str255) : OSErr;
RstFLock	Unlocks file	II, IV	FUNCTION RstFLock (fileName:Str255; vRefNum:INTEGER) : OSErr;
SetEOF	Sets logical end-of-file position	II, IV	FUNCTION SetEOF (refNum:INTEGER; logEOF:LONGINT) : OSErr;
SetFInfo	Sets Finder information for file	II, IV	FUNCTION SetFInfo (fileName:Str255; vRefNum:INTEGER; fndrInfo:FInfo) : OSErr;
SetFLock	Locks file	II, IV	FUNCTION SetFLock (fileName:Str255; vRefNum:INTEGER) : OSErr;
SetFPos	Sets filePos in open file	II, IV	FUNCTION SetFPos (refNum:INTEGER; posMode:INTEGER; posOff:LONGINT) : OSErr;
SetVol	Sets default volume	II, IV	FUNCTION SetVol (volName:StringPtr; vRefNum:INTEGER) : OSErr;
UnmountVol	Unmounts volume, calls FlushVol	II, IV	FUNCTION UnmountVol (volName:StringPtr; vRefNum:INTEGER) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V). Volume II used MFS; Volume IV used HFS system. Inside Macintosh pages II-97 to II-119 and IV-114 to IV-160 describe low-level routines not normally used by applications; pages V-386 to V-399 describe low-level shared environment calls.

Source: Inside Macintosh, pages II-89 to II-97, IV-106 to IV-114

See Also:
3.156. Operating System Calls
3.182. FInfo Data Structure
3.183. FXInfo Data Structure
3.184. DInfo Data Structure
3.185. DXInfo Data Structure
3.186. File Manager Result Codes
3.187. File Positioning Modes
3.188. File Information Flags
3.189. File Access Modes
3.190. File Access Permissions
3.191. ParamBlockRec Data Structure
3.192. HParamBlockRec Data Structure
3.193. CInfoPBRec Data Structure
3.194. CMovePBRec Data Structure
3.195. WDPBRec Data Structure
3.196. FCBPBRec Data Structure
3.197. Volume Control Blocks Data Structure
3.198. File Control Block Data Structure
3.199. Drive Queue (DrvQE1) Data Structure

3.182. FInfo Data Structure

Size	*Name*	*Comments*
OSType	fdType	4-byte file type string
OSType	fdCreator	4-byte file creator string
INTEGER	fdFlags	Flags (bit 13 = 1 if file has bundle, bit 14 = 1 if file invisible)
Point	fdLocation	Location of file in window (= 0 when creating file)
INTEGER	fdFldr	File's window (-3 = trash, -2 = desktop, 0 = disk window)

Notes: HFS adds FXInfo structure; see 3.183. FXInfo Data Structure.

Source: Inside Macintosh, pages II-84 to II-85

See Also: 3.181. File Manager Routines
3.183. FXInfo Data Structure

3.183. FXInfo Data Structure

Size	*Name*	*Comments*
INTEGER	fdIconID	ID of icon associated with file
8 bytes	fdUnused	RESERVED
INTEGER	fdComment	ID of comment for file
LONGINT	fdPutAway	ID of file's home directory

Source: Inside Macintosh, page IV-105

See Also: 3.181. File Manager Routines
3.182. FInfo Data Structure

3.184. DInfo Data Structure

Size	*Name*	*Comments*
Rect	frRect	Rectangle for folder's window
INTEGER	frFlags	Folder flags
Point	frLocation	Folder's location on desktop
INTEGER	frView	Folder's view

Source: Inside Macintosh, page IV-105

See Also: 3.181. File Manager Routines
3.185. DXInfo Data Structure

3.185. DXInfo Data Structure

Size	*Name*	*Comments*
Point	frScroll	Scroll position of folder window
LONGINT	frOpenChain	Directory ID chain of open folders
INTEGER	frUnused	RESERVED
INTEGER	frComment	ID of folder comment
LONGINT	frPutAway	ID of folder containing folder

Source: Inside Macintosh, page IV-106

See Also: 3.181. File Manager Routines
3.184. DInfo Data Structure

3.186. File Manager Result Codes

Value	*Constant*	*Comments*
-5021	RangeOverlap	Can't lock range with portions already locked
-5020	RangeNotLocked	Can't unlock range (user didn't lock range)
-5015	NoMoreLocks	Server cannot lock additional ranges
-5006	DenyConflict	Permission conflicts with open fork
-5000	AccessDenied	User doesn't have access to file/folder
-127	fsDSIntErr	Internal file system error
-124	VolGoneErr	Server volume disconnected
-123	wrgVolTypErr	Not a hierarchical volume
-122	badMovErr	Can't move into offspring
-121	tmwdoErr	Too many directories open
-120	dirNFErr	Directory was not found
-108	memFullErr	Heap zone not big enough
-61	wrPermErr	Permission doesn't allow write
-60	badMDBErr	Master directory block is bad
-59	fsRnErr	File could not be renamed
-58	extFSErr	External file system error
-57	noMacDskErr	No Mac-format directory on volume
-56	nsDrvErr	No match to drive number
-55	volOnLinErr	Volume is already on line
-54	permErr	Can't open locked file for write
-53	volOffLinErr	Volume specified is not on line
-52	gfpErr	Couldn't get filePos
-51	rfNumErr	Nonexistent access path
-50	paramErr	No default volume
-49	opWrErr	Multiple write requests to file
-48	dupFNErr	Named file already exists
-47	fBsyErr	File already open
-46	vLckdErr	Volume locked
-45	fLckdErr	File is locked
-44	wPrErr	Volume locked by hardware
-43	fnfErr	File wasn't found
-42	tmfoErr	Too many files are open
-40	posErr	Can't position before start of file
-39	eofErr	Logical EOF reached on read
-38	fnOpnErr	File is not open
-37	bdNamErr	File or volume name is bad
-36	ioErr	I/O error

(continued)

Table 3.186. Continued

Value	*Constant*	*Comments*
-35	nsvErr	Volume doesn't exist
-34	dskFulErr	Volume allocation blocks full
-33	dirFulErr	File directory is full
0	noErr	No error

Source: Inside Macintosh, pages II-138 to II-139, IV-201, V-387

See Also: 3.181. File Manager Routines

3.187. File Positioning Modes

Value	*Constant*	*Comments*
0	fsAtMark	Position at current mark
1	fsFromStart	Position offset from start of file
2	fsFromLEOF	Position offset from logical end of file
3	fsFromMark	Position offset from current mark
64	rdVerify *	

Notes: * Add this value to other modes if you want read verification.

Source: Inside Macintosh, page II-130

See Also: 3.181. File Manager Routines

3.188. File Information Flags

Value	*Constant*	*Comments*
-3	fTrash	File is in trash window
-2	fDesktop	File is on desktop
0	fDisk	File is in disk window
8192	fHasBundle	File has bundle bit set
16384	fInvisible	File is invisible (icon does not appear)

Source: Inside Macintosh, page II-130

See Also: 3.181. File Manager Routines

3.189. File Access Modes

Value	*Constant*	*Comments*
0	fsCurPerm	Request whatever is currently allowed
1	fsRdPerm	Request read-only status
2	fsWrPerm	Request write-only status
3	fsRdWrPerm	Request read and write status

Source: Inside Macintosh, page II-130

See Also: 3.181. File Manager Routines
3.190. File Access Permissions

3.190. File Access Permissions

Use	*Allowable Read Perms.*	*Allowable Write Perms.*	*Range Lock Allowed*	*File Access Mode*	*Deny-Mode Perm.*
Browse file	Read, deny write	Not allowed	No	fsRdPerm	Browsing
Single user access	Not allowed	Read, write, deny read, deny write	No	*	Exclusive **
Single write, shared read	Read	Read, write, deny write	Yes	*	Exclusive **
Shared read and write	Read	Read, write, deny none	Yes	fsRdWrShPerm	Shared

Notes:
- * No real equivalent; fsRdWrPerm, fsWrPerm, fsCurPerm translate into exclusive deny-mode permissions.
- ** Retried with browsing access if deny-mode permission fails.

Source: Inside Macintosh, pages V-377 to V-380

See Also: 3.181. File Manager Routines
3.189. File Access Modes

3.191. ParamBlockRec Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine
OSErr	ioResult	Result code
StringPtr	ioNamePtr	Name of volume or file
INTEGER	ioVRefNum	Ref number of volume or drive
22 bytes	ioParam	I/O routine parameters
54 bytes	fileParam	File information routine parameters
38 bytes	volumeParam	Volume info routine parameters
Varies	cntrlParam	Used by Device Manager only

INTEGER	ioRefNum
Signed byte	ioVersNum
Signed byte	ioPermssn
Ptr	ioMisc
Ptr	ioBuffer
LONGINT	ioReqCount
LONGINT	ioActCount
INTEGER	ioPosMode
LONGINT	ioPosOffset

INTEGER	ioFRefNum
Signed byte	ioFVersNum
Signed byte	filler1
INTEGER	ioFDirIndex
Byte	ioFlAttrib
Byte	ioFlVersNum
FInfo	ioFlFndrInfo
LONGINT	ioFlNum
INTEGER	ioFlStBlk
LONGINT	ioFlLgLen
LONGINT	ioFlPyLen
INTEGER	ioFlRStBlk
LONGINT	ioFlRLgLen
LONGINT	ioFlRPyLen
LONGINT	ioFlCrDat
LONGINT	ioFlMdDat

LONGINT	filler2
INTEGER	ioVolIndex
LONGINT	ioVCrData
LONGINT	ioVLsBkUp
INTEGER	ioVAtrb
INTEGER	ioVNmFls
INTEGER	ioVDirSt
INTEGER	ioVBlLn
INTEGER	ioVNmAlBlks
LONGINT	ioVABlkSiz
LONGINT	ioVClpSiz
INTEGER	ioAlBlSt
LONGINT	ioVNxtFNum
INTEGER	ioVFrBlk

Source: Inside Macintosh, pages II-98 to II-102, IV-116 to IV-127

See Also: 3.181. File Manager Routines
3.189. File Access Modes
3.190. File Access Permissions

3.192. HParamBlockRec Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine
OSErr	ioResult	Result code
StringPtr	ioNamePtr	Name of volume or file
INTEGER	ioVRefNum	Ref number of volume or drive
22 bytes	ioParam	I/O routine parameters
54 bytes	fileParam	File information routine parameters
38 bytes	volumeParam	Volume info routine parameters

LONGINT	filler2
INTEGER	ioVolIndex
LONGINT	ioVCrDate
LONGINT	ioVLsMod
INTEGER	ioVAtrb
INTEGER	ioVNmFls
INTEGER	ioVBitMap
INTEGER	ioAllocPtr
INTEGER	ioVNmAlBlks
LONGINT	ioVAlBlkSiz
LONGINT	ioVClpSiz
INTEGER	ioAlBlSt
LONGINT	ioVNxtCNID
INTEGER	ioVFrBlk
INTEGER	ioVSigWord
INTEGER	ioVDrvInfo
INTEGER	ioVDRefNum
INTEGER	ioVFSID
LONGINT	ioVBkUp
INTEGER	ioVSegNum
LONGINT	ioVWrCnt
LONGINT	ioVFilCnt
LONGINT	ioVDirCnt
8 LONGINTS	ioVFndrInfo

INTEGER	ioFRefNum
Signed byte	ioFVersNum
Signed byte	filler1
INTEGER	ioFDirIndex
Signed byte	ioFlAttrib
Signed byte	ioFlVersNum
FInfo	ioFlFndrInfo
LONGINT	ioDirID
INTEGER	ioFlStBlk
LONGINT	ioFlLgLen
LONGINT	ioFlPyLen
INTEGER	ioFlRStBlk
LONGINT	ioFlRLgLen
LONGINT	ioFlRPyLen
LONGINT	ioFlCrDat
LONGINT	ioFlMdDat

INTEGER	ioRefNum
Signed byte	ioVersNum
Signed byte	ioPermssn
Ptr	ioMisc
Ptr	ioBuffer
LONGINT	ioReqCount
LONGINT	ioActCount
INTEGER	ioPosMode
LONGINT	ioPosOffset

Source: Inside Macintosh, pages IV-116 to IV-127

See Also: 3.181. File Manager Routines
3.189. File Access Modes
3.190. File Access Permissions
3.191. ParamBlockRec Data Structure

3.193. CInfoPBRec Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine
OSErr	ioResult	Result code
StringPtr	ioNamePtr	Name of file or volume
INTEGER	ioVRefNum	Ref number of volume or drive
INTEGER	ioFRefNum	Ref number of file
Signed byte	ioFVersNum	Version number of file
Signed byte	filler1	NOT USED
INTEGER	ioFDirIndex	Index
Signed byte	ioFlAttrib	File attributes *
Signed byte	filler2	NOT USED
Varies		Depends on value of CInfoType:

----->

If CInfoType = hFileInfo

FInfo	ioFIFndrInfo	Finder info
LONGINT	ioDirID	Dir ID/file number
INTEGER	ioFlStBlk	1st alloc block of data fork
LONGINT	ioFlLgLen	Logical EOF data fork
LONGINT	ioFlPyLen	Phyiscal EOF data fork
INTEGER	ioFlRStBlk	1st alloc block res fork
LONGINT	ioFlRLgLen	Logical EOF res fork
LONGINT	ioFlRPyLen	Physical EOF res fork
LONGINT	ioFlCrDat	Creation date/time
LONGINT	ioFlMdDat	Modification date/time
LONGINT	ioFlBkDat	Backup date/time
FXInfo	ioFlXFndrInfo	Additional finder info
LONGINT	ioFlParID	Parent directory ID
LONGINT	ioFlClpSiz	Clump size

or

If CInfoType = dirInfo

DInfo	ioDrUsrWds	Finder info
LONGINT	ioDrDirID	Directory ID
INTEGER	ioDrNmFls	Number files in directory
9 INTEGERS	filler3	NOT USED
LONGINT	ioDrCrDat	Creation date/time
LONGINT	ioDrMdDat	Modification date/time
LONGINT	ioDrBkDat	Backup date/time
DXInfo	ioDrFndrInfo	Additional finder info
LONGINT	ioDrParID	Parent directory ID

Notes: * Bit 0 = file lock, 2 = resource fork open, 3 = data fork open, 4 = directory, 7 = file open (either fork).

Source: Inside Macintosh, pages IV-117 to IV-126

See Also: 3.181. File Manager Routines
3.182. FInfo Data Structure
3.183. FXInfo Data Structure
3.184. DInfo Data Structure
3.185. DXInfo Data Structure
3.186. File Manager Result Codes

3.194. CMovePBRec Data Structure

Size	*Name*	*Comments*
LONGINT	filler1	NOT USED
StringPtr	ioNewName	New directory name
LONGINT	filler2	NOT USED
LONGINT	ioNewDirID	ID of new directory
2 LONGINTS	filler3	NOT USED
LONGINT	ioDirID	ID of current directory

Notes: Added to standard ParamBlock when using CatMove.

Source: Inside Macintosh, page IV-127

See Also: 3.181. File Manager Routines
3.191. ParamBlockRec Data Structure

3.195. WDPBRec Data Structure

Size	*Name*	*Comments*
INTEGER	filler1	NOT USED
INTEGER	ioWDIndex	Index
LONGINT	ioWDProcID	User ID of working directory
INTEGER	ioWDVRefNum	Volume ref number of working directory
7 INTEGERS	filler2	NOT USED
LONGINT	ioWDDirID	Directory ID of working directory

Notes: Added to standard ParamBlock when using routines that open/close/get information about working directories.

Source: Inside Macintosh, page IV-127

See Also: 3.181. File Manager Routines
3.191. ParamBlockRec Data Structure

3.196. FCBPBRec Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine
OSErr	ioResult	Result code
StringPtr	ioNamePtr	Name of file or volume
INTEGER	ioVRefNum	Ref number of volume or drive
INTEGER	ioRefNum	Ref number of file
INTEGER	filler	NOT USED
LONGINT	ioFCBIndx	Index
LONGINT	ioFCBFlNm	File number
INTEGER	ioFCBFlags	Flags byte
INTEGER	ioFCBStBlk	Starting allocation block
LONGINT	ioFCBEOF	Logical end of file
LONGINT	ioFCBPLen	Physical end of file
LONGINT	ioFCBCrPs	Current mark position
INTEGER	ioFCBVRefNum	Volume reference number
LONGINT	ioFCBClpSiz	Clump size
LONGINT	ioFCBParID	Parent directory ID

Source: Inside Macintosh, pages IV-117, IV-179

See Also: 3.181. File Manager Routines
3.186. File Manager Result Codes

3.197. Volume Control Blocks Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	vcbFlags	Bit 15 = 1 if volume dirty (changed)
INTEGER	vcbSigWord	Type of directory structure ($4244 = HFS, $D2D7 = MFS)
LONGINT	vcbCrDate	Creation date and time
LONGINT	vcbLsMod	Last modification date and time
INTEGER	vcbAtrb	Volume attributes
INTEGER	vcbNmFls	Number of files in directory
INTEGER	vcbVBMSt	First block of volume's bitmap
INTEGER	vcbAllocPtr	USED INTERNALLY
INTEGER	vcbNmAlBlks	Number of allocation blocks in volume
LONGINT	vcbAlBlkSiz	Allocation block size for volume
LONGINT	vcbClpSiz	Default clump size
INTEGER	vcbAlBlSt	First block number in block map
LONGINT	vcbNxtCNID	Next unused directory of file ID number
INTEGER	vcbFreeBks	Number of unused allocation blocks on volume
27 bytes	vcbVN	Volume name
INTEGER	vcbDrvNum	Drive number

(continued)

Table 3.197. Continued

Size	*Name*	*Comments*
INTEGER	vcbDRefNum	Driver reference number
INTEGER	vcbFSID	File system ID
INTEGER	vcbVRefNum	Volume reference number
Ptr	vcbMAdr	Pointer to block map
Ptr	vcbBufAdr	Pointer to volume buffer
INTEGER	vcbMLen	Size of block map, in bytes
INTEGER	vcbDirIndex	USED INTERNALLY
INTEGER	vcbDirBlk	USED INTERNALLY
LONGINT	vcbVolBkUp	Last backup date and time
INTEGER	vcbVSeqNum	USED INTERNALLY
LONGINT	vcbWrCnt	Volume write count
LONGINT	vcbXTClpSiz	Extents tree file clump size
LONGINT	vcbCTClpSiz	Catalog tree file clump size
INTEGER	vcbNmRtDirs	Number of directories in root directory
LONGINT	vcbFilCnt	Number of files in entire volume
LONGINT	vcbDirCnt	Number of directories in entire volume
8 LONGINTS	vcbFndrInfo	Finder information
INTEGER	vcbVCSize	USED INTERNALLY
INTEGER	vcbVBMCSiz	USED INTERNALLY
INTEGER	vcbCtlCSiz	USED INTERNALLY
INTEGER	vcbXTAlBks	Extents tree file size, in blocks
INTEGER	vcbCTAlBks	Catalog tree file size, in blocks
INTEGER	vcbXTRef	Extents tree file path reference number
INTEGER	vcbCTRef	Catalog tree file path reference number
Ptr	vcbCtlBuf	Pointer to extents/catalog tree caches
LONGINT	vcbDirIDM	Last directory searched
INTEGER	vcbOffsM	Offspring index, at last search

Source: Inside Macintosh, pages II-125, IV-176 to IV-178

See Also: 3.181. File Manager Routines

3.198. File Control Block Data Structure

Size	*Name*	*Comments*
Long word	fcbFlNum	File number
Byte	fcbMdRByt	Flag byte
Byte	fcbTypByt	Version number
Word	fcbSBlk	First allocation block for file
Long word	fcbEOF	Logical end of file
Long word	fcbPLen	Physical end of file
Long word	fcbCrPs	Current position mark
Ptr	fcbVPtr	Pointer to volume control block
Ptr	fcbBfAdr	Pointer to access path buffer
Word	fcbFlPos	USED INTERNALLY
Long word	fcbClmpSize	File clump size
Long word	fcbBTCBPtr	Pointer to B*-tree control block
12 bytes	fcbExtRec	First three file extents
Long word	fcbFType	4-character file type string
Long word	fcbCatPos	USED INTERNALLY
Long word	fcbDirID	Parent ID for file
Varies	fcbCName	File name

Source: Inside Macintosh, pages IV-180 to IV-181

See Also: 3.181. File Manager Routines
3.197. Volume Control Blocks Data Structure

3.199. Drive Queue (DrvQE1) Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	dQDrive	Drive number
INTEGER	dQRefNum	Driver reference number
INTEGER	dQFSID	File system ID
INTEGER	dQDrvSz	Size of drive, in logical blocks
INTEGER	dQDrvSz2	Additional field for drive size

Source: Inside Macintosh, page IV-181

See Also: 3.181. File Manager Routines

3.200. Device Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
CloseDriver	Closes device driver	II	FUNCTION CloseDriver (refNum:INTEGER) : OSErr;
Control	Sends control info to driver	II	FUNCTION Control (refNum:INTEGER; csCode:INTEGER; csParamPtr:Ptr) : OSErr;
Device	Chooser communication with device	IV	FUNCTION Device (message,caller:INTEGER; objName,zoneName:StringPtr; p1,p2:LONGINT) : OSErr;
FSRead	Attempts to read from driver	II	FUNCTION FSRead (refNum:INTEGER; VAR count:LONGINT; buffPtr:Ptr) : OSErr;
FSWrite	Attempts to write to driver	II	FUNCTION FSWrite (refNum:INTEGER; VAR count:LONGINT; buffPtr:Ptr) : OSErr;
KillIO	Ends all current/pending I/O for driver	II	FUNCTION KillIO (refNum:INTEGER) : OSErr;
OpenDriver	Returns ref number to opened driver	II	FUNCTION OpenDriver (name:Str255; VAR refNum:INTEGER) : OSErr;
OpenSlot	Opens device in NuBus slot	V	FUNCTION OpenSlot (paramBlock:paramBlkPtr; async:BOOLEAN) : OSErr;
PBClose	Low level close driver routine	II	FUNCTION PBClose (paramBlock:ParmBlkPtr; async:BOOLEAN) : OSErr;
PBControl	Low level send control info	II	FUNCTION PBControl (paramBlock:ParmBlkPtr; async:BOOLEAN) : OSErr;
PBKillIO	Low level end current/pending I/O	II	FUNCTION PBKillIO (paramBlock:ParmBlkPtr; async: BOOLEAN) : OSErr;
PBOpen	Low level open driver routine	II	FUNCTION PBOpen (paramBlock:ParmBlkPtr; async:BOOLEAN) : OSErr;
PBRead	Low level read from driver	II	FUNCTION PBRead (paramBlock:ParmBlkPtr; async:BOOLEAN) : OSErr;
PBStatus	Low level get driver status	II	FUNCTION PBStatus (paramBlock:ParmBlkPtr; async:BOOLEAN) : OSErr;
PBWrite	Low level write to driver	II	FUNCTION PRWrite (paramBlcok:ParmBlkPtr; async:BOOLEAN) : OSErr;
SIntInstall	Adds new element to interrupt queue	V	FUNCTION SIntInstall (sIntQElemPtr:SQElemPtr; theSlot:INTEGER) : OSErr;
SIntRemove	Removes element from interrupt queue	V	FUNCTION SIntRemove (sIntQElemPtr:SQElemPtr; theSlot:INTEGER) : OSErr;
Status	Returns status info about driver	II	FUNCTION Status (refNum:INTEGER; csCode:INTEGER; csParamPtr:Ptr) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-178 to II-187, IV-217 to IV-218, V-425 to V-428

See Also:
3.156. Operating System Calls
3.201. Device Manager Result Codes
3.202. ParamBlockRec Data Structure
3.203. Device Driver Structure
3.204. Device Driver Flags
3.205. DCtlEntry Data Structure
3.206. I/O Queue Record Data Structure
3.207. Device Unit Table
3.208. Device Access Modes
3.209. Standard Unit Assignments
3.210. Chooser Device ID Flags
3.211. Device Positioning Modes
3.212 to 3.217. Interrupts tables
3.218. to 3.220. Graphics devices tables
3.221. to 3.228. Sound driver tables

3.201. Device Manager Result Codes

Value	*Constant*	*Comments*
-27	abortErr	I/O request was ended by KillIO function
-21	badUnitErr	Reference number doesn't match unit table
-17	controlErr	Driver didn't understand control call
-26	dInstErr	Driver wasn't found in resource file
-25	dRemovErr	Cannot remove an open driver
0	noErr	No error
-28	notOpenErr	Driver isn't open
-23	openErr	Requested r/w permission doesn't match driver's
-19	readErr	Driver doesn't respond to read calls
-18	statusErr	Driver doesn't respond to status calls
-22	unitEmptyErr	Reference number is NIL handle in unit table
-20	writErr	Driver doesn't repond to write calls

Source: Inside Macintosh, page II-204

See Also: 3.200. Device Manager Routines
3.202. Device ParamBlockRec Data Structure

3.202. Device ParamBlockRec Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine
OSErr	ioResult	Result code
StringPtr	ioNamePtr	Name of volume or file
INTEGER	ioVRefNum	Ref number of volume or drive
22 bytes	ioParam	I/O routine parameters
54 bytes	fileParam	Used by File Manager
38 bytes	volumeParam	Used by File Manager
26 bytes	cntrlParam	Control paramaters

INTEGER	ioCRefNum
INTEGER	csCode
11 INTEGERS	csParam

INTEGER	ioRefNum
Signed byte	ioVersNum
Signed byte	ioPermssn
Ptr	ioMisc
Ptr	ioBuffer
LONGINT	ioReqCount
LONGINT	ioActCount
INTEGER	ioPosMode
LONGINT	ioPosOffset

Source: Inside Macintosh, pages II-181 to II-183

See Also: 3.181. to 3.194. File Manager tables
3.200. Device Manager Routines
3.208. Device Access Modes

3.203. Device Driver Structure

Size	*Name*	*Comments*
Word	drvrFlags	See 3.204.
Word	drvrDelay	Delay between actions, in ticks
Word	drvrEMask	Desk accessory event mask
Word	drvrMenu	ID of menu for driver, if any
Word	drvrOpen	Offset to open routine
Word	drvrPrime	Offset to prime routine
Word	drvrCtl	Offset to control routine
Word	drvrStatus	Offset to status routine
Word	drvrClose	Offset to close routine
Varies	drvrName	Pascal string containing driver's name
Varies	driver routines	Code constituting routines

Source: Inside Macintosh, pages II-188 to II-189

See Also: 3.200. Device Manager Routines
3.204. Device Driver Flags

3.204. Device Driver Flags

Bit Numbers

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Name*	*Values*
X								NOT USED	
	X							dNeedLock	1 = driver locked in memory when opened
		X						dNeedTime	1 = driver performs periodic action
			X					dNeedGoodBye	1 = call driver before application heap reinitialized
				X				dStatEnable	1 = driver responds to status call
					X			dCtlEnable	1 = driver responds to control call
						X		dWritEnable	1 = driver responds to write call
							X	dReadEnable	1 = driver responds to read call

Source: Inside Macintosh, page II-188

See Also: 3.200. Device Manager Routines
3.203. Device Driver Structure

3.205. DCtlEntry Data Structure

Size	*Name*	*Comments*
Ptr	dCtlDriver	Pointer to ROM driver; handle to driver in RAM
INTEGER	dCtlFlags	See 3.204.
QHdr	dCtlQHdr	Driver I/O queue header
LONGINT	dCtlPosition	Position mark for read/write calls
Handle	dCtlStorage	Private storage for RAM driver
INTEGER	dCtlRefNum	Reference number for driver
LONGINT	dCtlCurTicks	USED INTERNALLY
WindowPtr	dCtlWindow	Driver's window, if any
INTEGER	dCtlDelay	Time between periodic actions, in ticks
INTEGER	dCtlEMask	Desk accessory event mask
INTEGER	dCtlMenu	ID of menu associated with driver, if any

--->

Word	dCtlQueue	Queue flags
Ptr	dCtlQHead	First entry in driver's I/O queue
Ptr	dCtlQTail	Last entry in driver's I/O queue

Notes: AuxDCE defined by Inside Macintosh Volume V appends the following to structure:

Byte	dCtlSlot
Byte	dCtlSlotID
LONGINT	dCtlDevBase
LONGINT	RESERVED
Byte	dCtlExtDev
Byte	fillByte

Source: Inside Macintosh, pages II-190, II-206, V-425

See Also: 3.200. Device Manager Routines
3.204. Device Driver Flags

3.206. I/O Queue Record Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type, must always be ORD (ioQType)
INTEGER	ioTrap	Routine trap for Device Manager routine
Ptr	ioCmdAddr	Routine address for Device Manager routine
Varies		

Source: Inside Macintosh, page II-191

See Also: 3.200. Device Manager Routines
3.202. Device ParamBlockRec Data Structure

3.207. Device Unit Table

Size	*Name*	*Unit Number*
Dbl word	RESERVED	0
Dbl word	XL hard disk	1
Dbl word	Printer Driver	2
Dbl word	Sound Driver	3
Dbl word	Disk Driver	4
Dbl word	Serial Driver Port A Input	5
Dbl word	Serial Driver Port A Output	6
Dbl word	Serial Driver Port B Input	7
Dbl word	Serial Driver Port B Output	8
Dbl word	AppleTalk .MPP Driver	9
Dbl word	AppleTalk .ATP Driver	10
Dbl word	RESERVED	11
Dbl word	Calculator	12
Dbl word	Alarm Clock	13
Dbl word	Key Caps	14
Dbl word	Puzzle	15
Dbl word	Note Pad	16
Dbl word	Scrapbook	17
Dbl word	Control Panel	18
	NOT USED	19–31

Notes: Apple doesn't guarantee that these unit numbers will stay valid at all times because the System and Font/DA Mover may change values to resolve resource conflicts.

Source: Inside Macintosh, pages II-191 to II-192

See Also: 3.200. Device Manager Routines

3.208. Device Access Modes

Value	*Constant*	*Comments*
0	fsCurPerm	Whatever is currently allowed
1	fsRdPerm	Read-only permission
2	fsWrPerm	Write-only permission
3	fsRdWrPerm	Read/write permission

Source: Inside Macintosh, page II-201

See Also: 3.200. Device Manager Routines
3.202. Device ParamBlockRec Data Structure

3.209. Standard Unit Assignments

Value	*Assignment*
0–11	System devices (see 3.207.)
12–26	DAs in System file
27–31	DAs in application files
32–39	SCSI drivers 0 through 7
40–47	RESERVED

Source: Inside Macintosh, page IV-215

See Also: 3.207. Device Unit Table

3.210. Chooser Device ID Flags

Bit Numbers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Description
X																1 = AppleTalk device
	X	X														RESERVED *
			X													1 = can have multiple instances selected
				X												1 = uses left button
					X											1 = uses right button
						X										1 = no saved zone name
							X									1 = uses actual zone names
								X	X	X	X	X	X	X		RESERVED *
															X	1 = accepts newSel message

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Description
X																1 = accepts fillList message
	X															1 = accepts getSel message
		X														1 = accepts select message
			X													1 = accepts deselect message
				X												1 = accepts terminate message
					X	X	X	X	X	X	X	X	X	X	X	RESERVED *

Notes: * Should be cleared to zero (0).

Source: Inside Macintosh, page IV-217

See Also: Chooser information in Inside Macintosh, pages IV-216 to IV-222

3.211. Device Positioning Modes

Value	Constant	Comments
0	fsAtMark	Position at current mark
1	fsFromStart	Position offset to beginning of medium
3	fsFromMark	Position offset relative to current mark
64	rdVerify *	Read/write permission

Notes: * Value added to other modes to verify reads.

Source: Inside Macintosh, page II-201

See Also: 3.200. Device Manager Routines
3.202. Device ParamBlockRec Data Structure

3.212. Interrupt Priority Levels

Level	Interrupt
0	NONE
1	VIA (Synertek SY6522 Versatile Int Adapter)
2	SCC (Zilog Z8530 Serial Com Controller)
3	VIA and SCC simultaneously
4–7	Debugging Switch

Source: Inside Macintosh, page II-196

See Also: 3.213. Processor Priority Levels
3.214. Primary Interrupt Vector Table
3.215. Level 1 Secondary Interrupt Vector Table
3.216. Level 2 Secondary Interrupt Vector Table
3.217. Level 2 External/Status Interrupt Vector Table

3.213. Processor Priority Levels

Level	*Interrupt*
0	All interrupts
1	SCC and debugging interrupts only
2–6	Debugging interrupts only
7	No interrupts

Source: Inside Macintosh, page II-196

See Also: 3.212. Interrupt Priority Levels
3.214. Primary Interrupt Vector Table
3.215. Level 1 Secondary Interrupt Vector Table
3.216. Level 2 Secondary Interrupt Vector Table
3.217. Level 2 External/Status Interrupt Vector Table

3.214. Primary Interrupt Vector Table

Location	*Name*	*Vector to*
$64	autoInt1	Level 1 interrupt handler
$68	autoInt2	Level 2 interrupt handler
$6C	autoInt3	Level 3 interrupt handler
$70	autoInt4	Level 4 interrupt handler
$74	autoInt5	Level 5 interrupt handler
$78	autoInt6	Level 6 interrupt handler
$7C	autoInt7	Level 7 interrupt handler

Source: Inside Macintosh, page II-197

See Also: 3.212. Interrupt Priority Levels
3.213. Processor Priority Levels
3.215. Level 1 Secondary Interrupt Vector Table
3.216. Level 2 Secondary Interrupt Vector Table
3.217. Level 2 External/Status Interrupt Vector Table

3.215. Level 1 Secondary Interrupt Vector Table

Byte	*Interrupt*	*Source of Interrupt*
0	One-second interrupt	CA2 control line on VIA
4	Vertical retrace interrupt	CA1 control line on VIA
8	Shift-register interrupt	Shift register on VIA
12	NOT USED	
16	NOT USED	
20	T2 Timer (disk driver)	Timer 2 on VIA
24	T1 Timer (sound driver)	Timer 1 on VIA
28	NOT USED	

Source: Inside Macintosh, page II-197

See Also: 3.212. Interrupt Priority Levels
3.213. Processor Priority Levels
3.214. Primary Interrupt Vector Table
3.216. Level 2 Secondary Interrupt Vector Table
3.217. Level 2 External/Status Interrupt Vector Table

3.216. Level 2 Secondary Interrupt Vector Table

Byte	*Interrupt*	*Priority*	*Source of Interrupt*
0	Transmit buffer empty (channel B)	5	
4	External/status change (channel B)	6	Mouse vertical movement
8	Receive character available (channel B)	4	
12	Special receive condition (channel B)		
16	Transmit buffer empty (channel A)	2	
20	External/status change (channel A)	3	Mouse horizontal movement
24	Receive character available (channel A)	1	
28	Special receive condition (channel A)		

Source: Inside Macintosh, page II-198

See Also: 3.212. Interrupt Priority Levels
3.213. Processor Priority Levels
3.214. Primary Interrupt Vector Table
3.215. Level 1 Secondary Interrupt Vector Table
3.217. Level 2 External/Status Interrupt Vector Table

3.217. Level 2 External/Status Interrupt Vector Table

Byte	*Interrupt*
0	Communications interrupt (channel B)
4	Mouse vertical interrupt
8	Communications interrupt (channel A)
12	Mouse horizontal interrupt

Source: Inside Macintosh, page II-199

See Also: 3.212. Interrupt Priority Levels
3.213. Processor Priority Levels
3.214. Primary Interrupt Vector Table
3.215. Level 1 Secondary Interrupt Vector Table
3.216. Level 2 Secondary Interrupt Vector Table

3.218. Graphics Devices Routines

Name	*Description*	*Defined **	*Pascal Definition*
DisposGDevice	Removes gDevice	V	FUNCTION DisposGDevice : GDHandle;
GetDeviceList	Returns handle to first device in list	V	FUNCTION GetDeviceList : GDHandle;
GetGDevice	Returns handle to current gDevice	V	FUNCTION GetGDevice : GDHandle;
GetMainDevice	Returns handle of device with menu bar	V	FUNCTIN GetMainDevice : GDHandle;
GetMaxDevice	Returns handle to deepest intersecting device	V	FUNCTION GetMaxDevice (globalRect:Rect) : GDHandle;
GetNextDevice	Returns handle of next device in list	V	FUNCTION GetNextDevice (gdh:GDHandle) : GDHandle;
InitGDevice	Sets gdRefNum to specified mode	V	PROCEDURE InitGDevice (gdRefNum:INTEGER; mode:LONGINT; gdh:GDHandle);
NewGDevice	Returns handle to new gDevice structure	V	FUNCTION NewGDevice (refNum:INTEGER; mode:LONGINT) : GDHandle;
SetDeviceAttribute	Sets device attributes	V	PROCEDURE SetDeviceAttribute (gdh:GDHandle; attribute:INTEGER; value:BOOLEAN);
SetGDevice	Sets current gDevice	V	PROCEDURE SetGDevice (gdh:GDHandle);
TestDeviceAttribute	Returns setting for attribute	V	FUNCTION TestDeviceAttribute (curDevice:GDHandle; attribute:INTEGER) : BOOLEAN;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-122 to V-125

See Also: 3.156. Operating System Calls
3.219. gDevice Data Structure
3.220. Graphics Devices Attribute Types

3.219. gDevice Data Structure

Size	*Name*	*Comments*
INTEGER	gdRefNum	Driver reference number
INTEGER	gdID	Search procedure's client ID
INTEGER	gdType	Device type *
ITabHandle	gdITable	Inverse table
INTEGER	gdResPref	Preferred resolution
SProcHndl	gdSearchProc	Search procedure list
CProcHndl	gdCompProc	Complement procedure list
INTEGER	gdFlags	gDevice flags (see 3.220.)
PixMapHandle	gdPMap	Displayed image's pixel map
LONGINT	gdRefCon	Reference value
GDHandle	gdNextGD	Next gDevice
Rect	gdRect	Global bounds for device
LONGINT	gdMode	Current mode for device
INTEGER	gdCCBytes	RowBytes for expanded cursor
INTEGER	gdCCDepth	Expanded cursor depth
Handle	gdCCXData	Expanded cursor data
Handle	gdCCXMask	Exanded cursor mask
LONGINT	gdReserved	RESERVED

Notes: * 0 = lookup table (CLUT), 1 = fixed table, 2 = direct values.

Source: Inside Macintosh, pages V-129 to V-130

See Also: 3.218. Graphics Devices Routines
3.220 Graphics Devices Attribute Types

3.220. Graphics Devices Attribute Types

Bit Numbers

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Used to determine*	*Values*
X																Active device	1 = device active, 0 = device inactive
	X															Device driver presence	1 = device has no driver
		X														Screen device presence	1 = device is a screen
			X													If device came from 'scrn'	1 = device initialized from 'scrn' resource
				X												Main screen	1 = device is main screen
					X											If device came from RAM	1 = device initialized from RAM
						X	X	X	X	X	X	X	X	X		NOT USED	
															X	Color	0 = mono, 1 = color

Source: Inside Macintosh, pages V-124 to V-125

See Also: 3.218. Graphics Devices Routines
3.219. gDevice Data Structure

3.221. Sound Driver Routines

Name	*Description*	*Defined **	*Pascal Definition*
StartSound	Produces sound in synth buffer	II	PROCEDURE StartSound (synthRec:Ptr; numBytes:LONGINT; completionRtn:ProcPtr);
StopSound	Stops production of sound	II	PROCEDURE StopSound;
SoundDone	Returns TRUE if no sound made or pending	II	FUNCTION SoundDone : BOOLEAN;
GetSoundVol	Returns current volume, in units 0–7	II	PROCEDURE GetSoundVol (VAR level:INTEGER);
SetSoundVol	Sets current volume, in units 0–7	II	PROCEDURE SetSoundVol (level:INTEGER);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-231 to II-233

See Also: 3.156. Operating System Calls
3.222. Synthesizer Types
3.223. SWSynthRec Data Structure
3.224. Note Values
3.225. Tone Data Structure
3.226. FTSynthRec Data Structure
3.227. FTSoundRec Data Structure
3.228. FFSynthRec Data Structure
3.240. to 3.244. Sound Manager tables

3.222. Sound Driver Synthesizer Types

Value	*Constants*	*Comments*
-1	swMode	Square-wave synthesizer
0	ffMode	Free-form synthesizer
1	ftMode	Four-tone synthesizer

Notes: Applies to Sound Driver only.

Source: Inside Macintosh, page II-225

See Also: 3.221. Sound Driver Routines
3.223. to 3.228. Synthesizer tables
3.241. Sound Manager Synthesizer Types

3.223. SWSynthRec Data Structure

Size	*Name*	*Comments*
INTEGER	Mode	Always swMode (-1)
Tones *	Triplets	Array of sounds

Notes: * Array of Tone structures; see 3.225.

Source: Inside Macintosh, page II-225

See Also: 3.221. Sound Driver Routines
3.225. Tone Data Structure

3.224. Note Values

Just-Tempered Scale

	Rate (for Four-Tone Synth)		*Count (for Square-Wave Synth)*	
Note	*Long*	*Fixed*	*Word*	*INTEGER*
C	$612B	0.37956	$5CBA	23738
C sharp	$667C	0.40033	$57EB	22507
D flat	$67A6	0.40488	$56EF	22255
D	$6D51	0.42702	$526D	21101
E dbl flat	$6E8F	0.43187	$5180	20864
D sharp	$71DF	0.44481	$4F21	20257
E flat	$749A	0.45547	$4D46	19782
E	$7976	0.47446	$4A2F	18991
F	$818F	0.50609	$458C	17804
F sharp	$88A5	0.53377	$41F0	16880
G flat	$8A32	0.53983	$4133	16691
G	$91C1	0.56935	$3DD1	15825
G sharp	$97D4	0.59308	$3B58	15192
A flat	$9B79	0.60732	$39F4	14836
A	$A1F3	0.63261	$37A3	14243
B dbl flat	$A3CA	0.63980	$3703	14083
A sharp	$AA0C	0.66425	$34FD	13565
B flat	$ACBF	0.67479	$3429	13353

(continued)

Table 3.224. Continued

	Rate (for Four-Tone Synth)		Count (for Square-Wave Synth)	
Note	Long	Fixed	Word	INTEGER
B	$B631	0.71169	$3174	12660
C	$C257	0.75914	$2E5D	11869
C sharp	$CCF8	0.80066	$2BF6	11254
D flat	$CF4C	0.80975	$2B77	11127
D	$DAA2	0.85403	$2936	10550
E dbl flat	$DD1D	0.86372	$28C0	10432
D sharp	$E3BE	0.88962	$2790	10128
E flat	$E935	0.91096	$26A3	9891
E	$F2ED	0.94893	$2517	9495
F	$1031E	1.01218	$22C6	8902
F sharp	$1114A	1.06754	$20F8	8440
G flat	$11465	1.07967	$2099	8345
G	$12382	1.13870	$1EE9	7913
G sharp	$12FA8	1.18616	$1DAC	7596
A flat	$136F1	1.21461	$1CFA	7418
A	$143E6	1.26523	$1BD1	7121
B dbl flat	$14794	1.27960	$1B81	7041
A sharp	$15418	1.32849	$1A7E	6782
B flat	$1597E	1.34958	$1A14	6676
B	$16C63	1.42339	$18BA	6330
C	$184AE	1.51828	$172F	5935
C sharp	$199EF	1.60130	$15FB	5627
D flat	$19E97	1.61949	$15BC	5564
D	$1B543	1.70805	$149B	5275
E dbl flat	$1BA3B	1.72746	$1460	5216
D sharp	$1C77B	1.77922	$13C8	5064
E flat	$1D26A	1.82193	$1351	4945
E	$1E5D9	1.89784	$128C	4748
F	$2063D	2.02437	$1163	4451
F sharp	$22294	2.13507	$107C	4220
G flat	$228C9	2.15932	$104D	4173
G	$24704	2.27740	$F74	3956
G sharp	$25F4F	2.37230	$ED6	3798
A flat	$26DE3	2.42924	$E7D	3709
A	$287CC	2.53046	$DE9	3561
B dbl flat	$28F28	2.55920	$DC1	3521
A sharp	$2A830	2.65698	$D3F	3391
B flat	$2B2FC	2.69916	$D0A	3338
B	$2D8C6	2.84677	$C5D	3165
Middle C	$3095B	3.03654	$B97	2967
C sharp	$333DE	3.20261	$AFD	2813
D flat	$33D2E	3.23898	$ADE	2782
D	$36A87	3.41612	$A4E	2638
E dbl flat	$37476	3.45493	$A30	2608
D sharp	$38EF7	3.55846	$9E4	2532
E flat	$3A4D4	3.64386	$9A9	2473
E	$3CBB2	3.79568	$946	2374
F	$40C7A	4.04874	$8B1	2225
F sharp	$44528	4.27014	$83E	2110
G flat	$45193	4.31865	$826	2086
G	$48E09	4.55482	$7BA	1978
G sharp	$4BE9F	4.74461	$76B	1899

(continued)

Table 3.224. Continued

	Rate (for Four-Tone Synth)		Count (for Square-Wave Synth)	
Note	Long	Fixed	Word	INTEGER
A flat	$4DBC5	4.85847	$73F	1855
A	$50F98	5.06091	$6F4	1780
B dbl flat	$51E4F	5.11839	$6E0	1760
A sharp	$55060	5.31396	$6A0	1696
B flat	$565F8	5.39832	$685	1669
B	$5B18B	5.69353	$62F	1583
C	$612B7	6.07310	$5CC	1484
C sharp	$667BD	6.40523	$57F	1407
D flat	$67A5C	6.47797	$56F	1391
D	$6D50D	6.83223	$527	1319
E dbl flat	$6E8EB	6.90984	$518	1304
D sharp	$71DEE	7.11691	$4F2	1266
E flat	$749A8	7.28772	$4D4	1236
E	$79764	7.59137	$4A3	1187
F	$818F3	8.09746	$459	1113
F sharp	$88A51	8.54030	$41F	1055
G flat	$8A326	8.63730	$413	1043
G	$91C12	9.10965	$3DD	989
G sharp	$97D3D	9.48921	$3B6	950
A flat	$9B78B	9.71696	$39F	927
A	$A1F30	10.12183	$37A	890
B dbl flat	$A3C9F	10.23680	$370	880
A sharp	$AA0BF	10.62791	$350	848
B flat	$ACBEF	10.79662	$343	835
B	$B6316	11.38705	$317	791
C	$C256D	12.14619	$2E6	742
C sharp	$CCF79	12.81044	$2BF	703
D flat	$CF4B9	12.95595	$2B7	695
D	$DAA1B	13.66447	$293	659
E dbl flat	$DD1D6	13.81967	$28C	652
D sharp	$EBDC	14.23383	$279	633
E flat	$E9350	14.57544	$26A	618
E	$F2EC8	15.18274	$251	593
F	$1031E7	16.19493	$22C	556
F sharp	$1114A1	17.08058	$210	528
G flat	$11464C	17.27460	$20A	522
G	$123824	18.21930	$1EF	495
G sharp	$12FA7B	18.97844	$1DB	475
A flat	$136F15	19.43391	$1D0	464
A	$143E61	20.24367	$1BD	445
B dbl flat	$14793D	20.47359	$1B8	440
A sharp	$15417F	21.25584	$1A8	424
B flat	$1597DE	21.59323	$1A1	417
B	$16C62D	22.77412	$18C	396
C	$184ADA	24.29239	$173	371
C sharp	$199EF2	25.62088	$160	352
D flat	$19E971	25.91188	$15C	348
D	$1B5436	27.32895	$14A	330
E dbl flat	$1BA3AC	27.63934	$146	326
D sharp	$1C77B8	28.46765	$13D	317
E flat	$1D26A0	29.15088	$135	309
E	$1E5D91	30.36549	$129	297

(continued)

Table 3.224. Continued

	Rate (for Four-Tone Synth)		Count (for Square-Wave Synth)	
Note	Long	Fixed	Word	INTEGER
F	$2063CE	32.38986	$116	278
F sharp	$222943	34.16118	$108	264
G flat	$228C97	34.54918	$105	261
G	$247047	36.43858	$F7	247
G sharp	$25F4F5	37.95686	$ED	237
A flat	$26DE2A	38.86783	$E8	232
A	$287CC1	40.48732	$DF	223
B dbl flat	$28F27A	40.94717	$DC	220
A sharp	$2A82FE	42.51169	$D4	212
B flat	$2B2FBD	43.18648	$D1	209
B	$2D8C59	45.54823	$C6	198

Equal-Tempered Scale

	Rate (for Four-Tone Synth)		Count (for Square-Wave Synth)	
Note	Long	Fixed	Word	INTEGER
C	$604C	0.37616	$5D92	23954
C sharp	$6606	0.39853	$5851	22609
D	$6C17	0.42223	$535C	21340
D sharp	$7284	0.44733	$4EAF	20143
E	$7953	0.47392	$4A44	19012
F	$808A	0.50211	$4619	17945
F sharp	$882F	0.53197	$422A	16938
G	$9048	0.56360	$3E73	15987
G sharp	$98DC	0.59711	$3AF2	15090
A	$A1F3	0.63261	$37A3	14243
A sharp	$AB94	0.67023	$3484	13444
B	$B5C8	0.71008	$3191	12689
C	$C097	0.75230	$2EC9	11977
C sharp	$CC0B	0.79704	$2C29	11305
D	$D82D	0.84444	$29AE	10670
D sharp	$E508	0.89465	$2757	10071
E	$F2A6	0.94785	$2522	9506
F	$10114	1.00421	$230C	8972
F sharp	$1105D	1.06392	$2115	8469
G	$12090	1.12720	$1F3A	7994
G sharp	$131B8	1.19421	$1D79	7545
A	$143E6	1.26523	$1BD1	7121
A sharp	$15729	1.34047	$1A42	6722
B	$16B90	1.42017	$18C8	6344
C	$1812F	1.50462	$1764	5988
C sharp	$19816	1.59409	$1614	5652
D	$1B05A	1.68887	$14D7	5335
D sharp	$1CA10	1.78931	$13AC	5036
E	$1E54D	1.89571	$1291	4753
F	$20228	2.00842	$1186	4486
F sharp	$220BB	2.12785	$108A	4234
G	$2411F	2.25438	$F9D	3997
G sharp	$26370	2.38843	$EBC	3772
A	$287CC	2.53046	$DE9	3561
A sharp	$2AE51	2.68092	$D21	3361
B	$2D721	2.84035	$C64	3172
Middle C	$3025D	3.00923	$BB2	2994

(continued)

Table 3.224. Continued

	Rate (for Four-Tone Synth)		Count (for Square-Wave Synth)	
Note	Long	Fixed	Word	INTEGER
C sharp	$3302C	3.18817	$B0A	2826
D	$360B5	3.37776	$A6C	2668
D sharp	$29420	3.57861	$9D6	2518
E	$3CA99	3.79140	$949	2377
F	$40450	4.01685	$8C3	2243
F sharp	$44176	4.25571	$845	2117
G	$4823E	4.50876	$7CE	1998
G sharp	$4C6E1	4.77687	$75E	1886
A	$50F98	5.06091	$6F4	1780
A sharp	$55CA2	5.36185	$690	1680
B	$5AE41	5.68068	$632	1586
C	$604BB	6.01848	$5D9	1497
C sharp	$66059	6.37636	$585	1413
D	$6C169	6.75551	$536	1334
D sharp	$7283F	7.15721	$4EB	1259
E	$79533	7.58281	$4A4	1188
F	$808A1	8.03371	$462	1122
F sharp	$882EC	8.51141	$423	1059
G	$9047D	9.01753	$3E7	999
G sharp	$98DC2	9.55374	$3AF	943
A	$A1F30	10.12183	$37A	890
A sharp	$AB945	10.72371	$348	840
B	$B5C83	11.36137	$319	793
C	$C0976	12.03696	$2ED	749
C sharp	$CC0B1	12.75270	$2C3	707
D	$D82D2	13.51102	$29B	667
D sharp	$E507E	14.31442	$275	629
E	$F2A65	15.16560	$252	594
F	$101141	16.06740	$231	561
F sharp	$1105D8	17.02283	$211	529
G	$1208F9	18.03505	$1F4	500
G sharp	$131B83	19.10747	$1D8	472
A	$143E61	20.24367	$1BD	445
A sharp	$15728A	21.44742	$1A4	420
B	$16B906	22.72275	$18D	397
C	$1812EB	24.07390	$176	374
C sharp	$198163	25.50542	$161	535
D	$1B05A5	27.02205	$14D	333
D sharp	$1CA0FD	28.62886	$13B	315
E	$1E54CB	30.33122	$129	297
F	$202283	32.13481	$118	280
F sharp	$220BAF	34.04564	$109	265
G	$2411F2	36.07010	$FA	250
G sharp	$263706	38.21494	$EC	236
A	$287CC1	40.48732	$DF	223
A sharp	$2AE513	42.89482	$D2	210
B	$2D720B	45.44548	$C6	198

Source: Inside Macintosh, pages II-237 to II-241

See Also: 3.223. SWSynthRec Data Structure
3.226. FTSynthRec Data Structure
3.227. FTSoundRec Data Structure

3.225. Tone Data Structure

Size	*Name*	*Comments*
INTEGER	count	Frequency of tone
INTEGER	amplitude	Amplitude (0–255)
INTEGER	duration	Duration, in ticks

Source: Inside Macintosh, page II-225

See Also: 3.221. Sound Driver Routines
3.223. SWSynthRec Data Structure

3.226. FTSynthRec Data Structure

Size	*Name*	*Comments*
INTEGER	mode	Always 1
FTSndRecPtr	sndRec	See 3.227.

Source: Inside Macintosh, page II-227

See Also: 3.221. Sound Driver Routines
3.227. FTSoundRec Data Structure

3.227. FTSoundRec Data Structure

Size	*Name*	*Comments*
INTEGER	duration	Length of sound, in ticks
Fixed	sound1Rate	Cycle rate
LONGINT	sound1Phase	Byte offset
Fixed	sound2Rate	Cycle rate
LONGINT	sound2Phase	Byte offset
Fixed	sound3Rate	Cycle rate
LONGINT	sound3Phase	Byte offset
Fixed	sound4Rate	Cycle rate
LONGINT	sound4Phase	Byte offset
WavePtr	sound1Wave	Waveform for sound 1
WavePtr	sound2Wave	Waveform for sound 2
WavePtr	sound3Wave	Waveform for sound 3
WavePtr	sound4Wave	Waveform for sound 4

Source: Inside Macintosh, page II-227

See Also: 3.221. Sound Driver Routines
3.226. FTSynthRec Data Structure

3.228. FFSynthRec Data Structure

Size	*Name*	*Comments*
INTEGER	mode	Always 0
Fixed	count	Sampling factor
FreeWave	waveBytes	Description of waveform

Source: Inside Macintosh, page II-228

See Also: 3.221. Sound Driver Routines

3.229. Disk Driver Routines

Name	*Description*	*Defined **	*Pascal Definition*
DiskEject	Ejects disk from drive	II	FUNCTION DiskEject (drvNum:INTEGER) : OSErr;
SetTagBuffer	Changes info in file tags	II	FUNCTION SetTagBuffer (buffPtr:Ptr) : OSErr;
DriveStatus	Returns DrvSts record	II	FUNCTION DriveStatus (drvNum:INTEGER; VAR status:DrvSts) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-214 to II-215

See Also: 3.156. Operating System Calls
3.230. Driver Positioning Modes
3.231. DrvSts Data Structure
3.232. Disk Driver Result Codes
3.233. Advanced Control Calls

3.230. Driver Positioning Modes

Value	*Constant*	*Comments*
0	fsAtMark	At current sector
1	fsFromStart	Offset from first sector
3	fsFromMark	Offset from current sector

Source: Inside Macintosh, page II-212

See Also: 3.229. Disk Driver Routines

3.231. DrvSts Data Structure

Size	*Name*	*Values/Comments*
INTEGER	track	Current track
Signed byte	writeProt	If bit 7 = 1, volume is locked
Signed byte	diskInPlace	0 = no disk in drive; 1, 2 = disk in drive; -4, -1 = disk ejected in last 1.5 secs
Signed byte	installed	1; 0 = drive might be installed; -1 = drive not installed
Signed byte	sides	If bit 7 = 0, drive is single-sided
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	RESERVED
INTEGER	dQDrive	Drive number
INTEGER	dQRefNum	Reference number for driver
INTEGER	dQFSID	File system identifier
Signed byte	twoSideFmt	If -1, disk is double-sided
Signed byte	needsFlush	RESERVED
INTEGER	diskErrs	Count of errors

Source: Inside Macintosh, page II-215

See Also: 3.229. Disk Driver Routines

3.232. Disk Driver Result Codes

Value	*Constant*	*Comments*
0	noErr	No error
-44	wPrErr	Volume write-protected in hardware
-50	paramErr	Unknown positioning mode
-56	nsDrvErr	No such drive
-64	noDriveErr	No such disk connected
-64	lastDskErr	Last low-level disk error
-65	offLinErr	Drive does not have disk in it
-66	noNybErr	Disk may be blank—no data found
-67	noAdrMkErr	Couldn't find address mark
-68	dataVerErr	Failed to verify read
-69	badCksmErr	Encountered bad checksum
-70	badBtSlpErr	Encountered bad address mark
-71	noDtaMkErr	Couldn't find data mark
-72	badDCksum	Encountered bad data mark checksum
-73	badDBtSlp	Encountered bad data mark
-74	wrUnderrun	Write underrun error
-75	cantStepErr	Couldn't step drive
-76	tk0BadErr	Couldn't find track 0
-77	initIWMErr	Couldn't initialize disk controller
-78	twoSideErr	Attempted to read second side of single-sided disk
-79	spdAdjErr	Speed couldn't be adjusted correctly
-80	seekErr	Drive seek error
-81	sectNFErr	Couldn't find sector
-84	firstDskErr	First low-level disk error

Source: Inside Macintosh, pages II-217 to II-218

See Also: 3.229. Disk Driver Routines

3.233. Advanced Control Calls

csCode	*Description*	*csParam*
5	Verifies disk in the drive is formatted	None
6	Formats disk in drive	(INTEGER) 1 for singled-side, otherwise 0
9	Enable/disable track cache	(INTEGER) nonzero to enable, 0 to disable
21	Returns icon data structure for drive	(Ptr to Long)
22	Returns pointer to icon and mask for drive	(Ptr to Long)
23	Returns drive information	(Long) *

Notes: * One of following values in LO byte:
0 = no such drive
1 = unspecified drive
2 = 400K
3 = 800K
4–6 = RESERVED
7 = HD 20
HO byte bit flags:
Bit 8: 1 = external drive, 0 = internal
Bit 9: 1 = SCSI drive, 0 = IWM drive
Bit 10: 1 = fixed drive, 0 = removable media drive
Bit 11: 1 = secondary drive, 0 = primary drive

Source: Inside Macintosh, pages IV-223 to IV-224, V-470 to V-471

See Also: 3.229. Disk Driver Routines

3.234. SCSI Manager Routines

Name	*Description*	*Defined* *	*Pascal Definition*
SCSICmd	Sends SCSI command	IV	FUNCTION SCSICmd (buffer:Ptr; count:INTEGER) : OSErr;
SCSIComplete	Sets ticks for current command to complete	IV	FUNCTION SCSIComplete (VAR stat,message:INTEGER; wait:LONGINT) : OSErr;
SCSIGet	Arbitrates for use of SCSI bus	IV	FUNCTION SCSIGet : OSErr;
SCSIMsgIn	Gets message from SCSI device	V	FUNCTION SCSIMsgIn (VAR message:INTEGER) : OSErr;
SCSIMsgOut	Sends message to SCSI device	V	FUNCTION SCSIMsgOut (message:INTEGER) : OSErr;
SCSIRBlind	Reads from target to initiator without poll or wait	IV	FUNCTION SCSIRBling (tibPtr:Ptr) : OSErr;
SCSIRead	Reads from target to initiator	IV	FUNCTION SCSIRead (tibPtr:Ptr) : OSErr;
SCSIReset	Resets SCSI bus	IV	FUNCTION SCSIReset : OSErr;
SCSISelAtn	Selects SCSI device with attention line set	V	FUNCTION SCSISelAtn (targetID:INTEGER) : OSErr;
SCSISelect	Selects SCSI device (ID)	IV	FUNCTION SCSISelect (targetID:INTEGER) : OSErr;
SCSIStat	Returns SCSI control and status bits	IV	FUNCTION SCSIStat : INTEGER;
SCSIWBlind	Writes data from initiator to target without poll	IV	FUNCTION SCSIWBlind (tibPtr:Ptr) : OSErr;
SCSIWrite	Writes data from initiator to target	IV	FUNCTION SCSIWrite (tibPtr:Ptr) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages IV-289 to IV-291, V-575

See Also: 3.156. Operating System Calls
3.235. SCSI Operations
3.236. SCSI Control and Status Bits
3.237. SCSI Driver Descriptor Map
3.238. SCSI Device Partition Map
3.239. SCSI Result Codes

3.235. SCSI Operations

Value	*Constant*
0	scsiReset
1	scsiGet
2	scsiSelect
3	scsiCmd
4	scsiComplete
5	scsiRead
6	scsiWrite
7	NOT USED
8	scsiRBlind
9	scsiWBlind
10	scsiStat
11	scsiSelAtn
12	scsiMsgIn
13	scsiMsgOut

Notes: These values are used to select routines from assembly language; see source.

Source: Inside Macintosh, pages IV-289, V-575

See Also: 3.234. SCSI Manager Routines

3.236. SCSI Control and Status Bits

Bit Numbers

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Description*
X																End DMA
	X															DMA Request
		X														Parity Error
			X													Interrupt Request
				X												Phase Mat
					X											Busy Error
						X										Attention
							X									Acknowledge
								X								Reset
									X							Busy
										X						Request
											X					Message
												X				C/D
													X			I/O
														X		Select
															X	DBP

Source: Inside Macintosh, page II-291

See Also: 3.234. SCSI Manager Routines

3.237. SCSI Driver Descriptor Map

Size	*Name*	*Values/Comments*
Word	sbSig	Signature: always $4552
Word	sbBlkSize	Device block size, in bytes
Long word	sbBlkCount	Size of device, in blocks
Word	sbDevType	Type of device
Word	sbDevID	ID of device
Long word	sbData	NOT USED
Word	sbDrvrCount	Number of driver descriptors
Varies	dd	Driver descriptor table

----> *One of these for each driver:*

Long word	ddBlock
Word	ddSize
Word	ddType

Source: Inside Macintosh, pages II-292 to II-293

See Also: 3.234. SCSI Manager Routines

3.238. SCSI Device Partition Map

Size	*Name*	*Comments*
Word	pdSig	Signature: always $5453
Varies	pd	Partition data table

----> *One of these for each driver:*

Long word	pdStart
Long word	pdSize
Long word	pdFSID

Source: Inside Macintosh, pages II-292 to II-293

See Also: 3.234. SCSI Manager Routines

3.239. SCSI Result Codes

Value	*Constant*	*Meaning*
4	scBadParmsErr	Unrecognized instruction in block
2	scCommErr	SCSI protocol breakdown
6	scCompareErr	Data comparison error
5	scPhaseErr	Target/initiator don't agree on information to transfer
3	scArbNBErr	Bus busy, arbitration failed on SCSIGet
9	scBusTOErr	Bus timeout before data ready on blind read/write
10	scComplPhaseErr	Bus not in Status phase, SCSIComplete failed
7	scMgrBusyErr	Manager busy when SCSIGet called
8	scSequenceErr	Operation out of sequence

Source: Inside Macintosh, pages IV-294 , V-583

See Also: 3.234. SCSI Manager Routines

3.240. Sound Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
SndPlay	Plays 'snd ' resource	V	FUNCTION SndPlay (chan:SndChannelPtr; sndHdl:Handle; async:BOOLEAN) : OSErr;
SndNewChannel	Opens new channel for play	V	FUNCTION SndNewChannel (VAR chan:SndChannelPtr; synth:INTEGER; init:LONGINT; userRoutine:ProcPtr) : OSErr;
SndAddModifier	Adds modifier to open channel	V	FUNCTION SndAddModifier (chan:SndChannelPtr; modifier:ProcPtr; id:INTEGER; init:LONGINT) : OSErr;
SndDoCommand	Appends command to end of channel	V	FUNCTION SndDoCommand (chan:SndChannelPtr; cmd:SndCommand; noWait:BOOLEAN) : OSErr;
SndDoImmediate	Passes command directly to modifiers/synthesizers	V	FUNCTION SndDoImmediate (chan:SndChannelPtr; cmd:SndCommand) : OSErr;
SndControl	Sends command directly to modifiers/synthesizers	V	FUNCTION SndControl (id:INTEGER; VAR cmd:SndCommand) : OSErr;
SndDisposeChannel	Closes sound channel	V	FUNCTION SndDisposeChannel (chan:SndChannelPtr; quitNow:BOOLEAN) : OSErr;

Notes: • These commands replace old sound driver routines.
•* Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-477 to V-479

See Also: 3.156. Operating System Calls
3.221. Sound Driver Routines
3.241. Synthesizer Types
3.242. SndChannel Data Structure
3.243. ModifierStub Data Structure
3.244. Sound Manager Commands

3.241. Sound Manager Synthesizer Types

Value	*Constant*
1	noteSynth
3	waveTableSynth
5	sampledSynth
7	MIDISynthIn
9	MIDISynthOut

Notes: Applies to Sound Manager only.

Source: Inside Macintosh, page V-478

See Also: 3.221. Sound Driver Routines
3.222. Sound Driver Synthesizer Types
3.240. Sound Manager Routines
3.242. SndChannel Data Structure

3.242. SndChannel Data Structure

Size	*Name*	*Comments*
Ptr	nextChan	Pointer to next channel
Ptr	firstMod	Pointer to first modifier
ProcPtr	callBack	Pointer to call back procs
LONGINT	userInfo	User-defined
Time	wait	USED INTERNALLY
SndCommand	cmdInProg	USED INTERNALLY
INTEGER	flags	USED INTERNALLY
INTEGER	qLength	Queue length, in commands
INTEGER	qHead	USED INTERNALLY
INTEGER	qTail	USED INTERNALLY
Varies	queue	Array of SndCommand

Source: Inside Macintosh, page V-481

See Also: 3.240. Sound Manager Routines
3.243. ModifierStub Data Structure
3.244. Sound Manager Commands

3.243. ModifierStub Data Structure

Size	*Name*	*Comments*
Ptr	nextStub	Pointer to next modifier
ProcPtr	code	Pointer to modifier's code
LONGINT	userInfo	User-defined
Time	count	USED INTERNALLY
Time	every	USED INTERNALLY
Signed byte	flags	USED INTERNALLY
Signed byte	hState	USED INTERNALLY

Source: Inside Macintosh, page V-482

See Also: 3.240. Sound Manager Routines
3.242. SndChannel Data Structure
3.244. Sound Manager Commands

3.244. Sound Manager Commands

SndCommand Data Structure

Size	*Name*	*Comments*
INTEGER	cmd	Command * **
INTEGER	param1	First parameter, if any
LONGINT	param2	Second parameter, if any

Notes: •* Sound commands are as follows:

Value	*Constant*	*Param1*	*Param2*
0	nullCmd	NOT USED (0)	NOT USED (0)
1	initCmd	NOT USED (0)	Init parameter (see source, page V-486)
2	freeCmd	NOT USED (0)	NOT USED (0)
3	quietCmd	NOT USED (0)	NOT USED (0)
4	flushCmd	NOT USED (0)	NOT USED (0)
10	waitCmd	Duration, in ticks	NOT USED (0)
11	pauseCmd	NOT USED (0)	NOT USED (0)
12	resumeCmd	NOT USED (0)	NOT USED (0)
13	callBackCmd	User-defined	User-defined
14	syncCmd	Count	Identifier
15	emptyCmd	NOT USED (0)	NOT USED (0)
20	tickleCmd	NOT USED (0)	NOT USED (0)
21	requestNextCmd	Count	NOT USED (0)
22	howOftenCmd	Period, in half milliseconds	Pointer to ModifierStub
23	wakeUpCmd	Period, in milliseconds	Pointer to ModifierStub
24	availableCmd	Status	Init parameter (see source, page V-486)
40	noteCmd	Duration, in half milliseconds	LO byte = 0–127 for equal-tempered notes MO bytes = frequency, LO byte = fraction HO byte = NOT USED
41	restCmd	Duration, in milliseconds	NOT USED (0)
42	freqCmd	NOT USED (0)	Frequency (see noteCmd, above)
43	ampCmd	Amplitude	NOT USED (0)
44	timbreCmd	Timbre (0 = sine, 255 = square)	NOT USED (0)
60	waveTableCmd	Length	Pointer to table
61	phaseCmd	Undocumented	Undocumented
80	soundCmd	NOT USED (0)	Pointer to sound
81	bufferCmd	NOT USED (0)	Pointer to buffer
82	rateCmd	NOT USED (0)	Rate
100	midiDataCmd	Undocumented	Undocumented

•** HO bit set indicates param2 is a pointer to additional data.

Source: Inside Macintosh, pages V-482 to V-483, V-486 to V-490

See Also: 3.240. Sound Manager Routines
3.242. SndChannel Data Structure
3.243. ModifierStub Data Structure

3.245. Serial Driver Routines

Name	*Description*	*Defined **	*Pascal Definition*
RAMSDClose	Closes RAM driver	II	PROCEDURE RAMSDClose (whichPort:SPortSel);
RAMSDOpen	Closes ROM driver, opens RAM driver	II	FUNCTION RAMSDOpen (whichPort:SPortSel) : OSErr;
SerClrBrk	Clears break mode	II	FUNCTION SerClrBrk (refNum:INTEGER) : OSErr;
SerGetBuf	Counts bytes in buffer	II	FUNCTION SerGetBuf (refNum:INTEGER; VAR count:LONGINT) : OSErr;
SerHShake	Sets handshake/control information	II	FUNCTION SerHShake (refNum:INTEGER; flags:SerShk) : OSErr;
SerReset	Resets/reinitializes driver	II	FUNCTION SerReset (refNum:INTEGER; serConfig:INTEGER) : OSErr;
SerSetBrk	Sets break mode	II	FUNCTION SerSetBrk (refNum:INTEGER) : OSErr;
SerSetBuf	Sets up new input buffer	II	FUNCTION SerSetBuf (refNum:INTEGER; serBPtr:Ptr; serBLen:INTEGER) : OSErr;
SerStatus	Returns status record	II	FUNCTION SerStatus (refNum:INTEGER; VAR serSta:SerStaRec) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-249 to II-254

See Also: 3.156. Operating System Calls
3.246. Serial Driver Reference Numbers
3.247. Driver Reset Word
3.248. SerShk Data Structure
3.249. SerStaRec Data Structure
3.250. Advanced Control Calls
3.251. Serial Driver Result Codes

3.246. Serial Driver Reference Numbers

Number	*Constant*	*Driver*
-6	.AIn	Modem port input
-7	.AOut	Modem port output
-8	.BIn	Printer port input
-9	.BOut	Printer port output

Source: Inside Macintosh, page II-248

See Also: 3.245. Serial Driver Routines

3.247. Driver Reset Word

Bit Numbers

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Use*
X	X															Stop bits *
		X	X													Parity **
				X	X											Data bits ***
						X	X	X	X	X	X	X	X	X	X	Baud rate ****

Notes:
- * 01 = 1 stop bit (stop10), 10 = 1.5 stop bits (stop15), 11 = 2 stop bits (stop20).
- ** 00 = no parity (noParity), 01 = odd parity (oddParity), 10 = no parity (noParity), 11 = even parity (evenParity).
- *** 00 = 5 data bits (data5), 01 = 7 data bits (data7), 10 = 6 data bits (data6), 11 = 8 data bits (data8).
- **** Baud rates are determined as follows:

Value	*Baud Rate*	*Constant*
380	300	baud300
189	600	baud600
94	1200	baud1200
62	1800	baud1800
46	2400	baud2400
30	3600	baud3600
22	4800	baud4800
14	7200	baud7200
10	9600	baud9600
4	19200	baud19200
0	57600	baud57600

Source: Inside Macintosh, pages II-250 to II-251

See Also: 3.245. Serial Driver Routines

3.248. SerShk Data Structure

Size	*Name*	*Comments*
Byte	fXOn	XOn/XOff output flag
Byte	fCTS	Hardware CTS flag
CHAR	xOn	XOn character
CHAR	xOff	XOff character
Byte	errs	Errors cause abort
Byte	evts	Status changes cause events
Byte	fInX	XOn/XOff input flag
Byte	null	NOT USED

Source: Inside Macintosh, page II-252

See Also: 3.245. Serial Driver Routines

3.249. SerStaRec Data Structure

Size	*Name*	*Comments*
Byte	cumErrs	Number of cumulative errors
Byte	xOffSent	XOff sent as input flow control
Byte	rdPend	Read-pending flag
Byte	wrPend	Write-pending flag
Byte	ctsHold	CTS flow control hold flag
Byte	xOffHold	XOff flow control hold flag

Source: Inside Macintosh, page II-253

See Also: 3.245. Serial Driver Routines

3.250. Advanced Control Calls

csCode	*Description*	*csParam*
13	Sets baud rate	baudRate (actual integer value)
14	Adds DTR handshake to SerHShake	serShk structure starts at csParam
16	Sets miscellaneous control options	Option *
17	Asserts DTR	
18	Negates DTR	
19	Incoming parity errors replaced with char	Replacement CHAR
20	Extended call 19	Replacement CHAR followed by alternate CHAR
21	Sets XOff unconditionally	
22	Clears XOff unconditionally	
23	Sends XOn character if last in was XOff	
24	Sends XOn char unconditionally	
25	Sends XOff if last in was XOn	
26	Sends XOff unconditionally	
27	Resets SCC channel	

Notes: * Bits 0–6 should be 0; bit 7 leaves DTR unchanged if set.

Source: Inside Macintosh, pages II-254 to II-255, IV-226

See Also: 3.245. Serial Driver Routines

3.251. Serial Driver Result Codes

Value	*Constant*	*Comments*
0	noErr	No error
-23	openErr	Attempt to open RAM driver failed
-97	portInUse	Port already in use, Open failed
-98	portNotCf	Port not configured correctly, Open failed
-108	memFullErr	Heap zone full

Source: Inside Macintosh, pages II-256, IV-228

See Also: 3.245. Serial Driver Routines

3.252. System Error Handler Routine

Name	*Description*	*Defined **	*Pascal Definition*
SysError	Generates system error	II	PROCEDURE SysError (errorCode:INTEGER);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-362 to II-363

See Also: 3.156. Operating System Calls
3.253. System Errors

3.253. System Errors

Value	*Constant* *	*Description*
1	dsBusError	Bus error
2	dsAddressErr	Address error
3	dsIllInstErr	Illegal instruction
4	dsZeroDivErr	Zero divide attempt
5	dsChkErr	Check exception generated
6	dsOvflowErr	TrapV exception generated
7	dsPrivErr	Privilege violation
8	dsTraceErr	Trace exception
9	dsLineAErr	Line 1010 exception generated
10	dsLineFErr	Line 1111 exception generated
11	dsMiscErr	Miscellaneous exception generated
12	dsCoreErr	Unimplemented core routine
13	dsIrqErr	Spurious interrupt generated
14	dsIOCoreErr	I/O system error
15	dsLoadErr	Segment Loader error
16	dsFPErr	Floating-point error
17	dsNoPackErr	Package 0 couldn't be loaded
18	dsNoPk1	Package 1 couldn't be loaded
19	dsNoPk2	Package 2 couldn't be loaded
20	dsNoPk3	Package 3 couldn't be loaded
21	dsNoPk4	Package 4 couldn't be loaded
22	dsNoPk5	Package 5 couldn't be loaded
23	dsNoPk6	Package 6 couldn't be loaded
24	dsNoPk7	Package 7 couldn't be loaded
25	dsMemFullErr	Requested block can't be allocated
26	dsBadLaunch	Segment Loader error
27	dsFSErr	File map destroyed
28	dsStkNHeap	Stack overflow
30	dsReinsert	"Please insert the disk" message
31	dsNotThe1	Incorrect disk inserted
33	negZcbFreeErr	ZcbFree is negative
84	menuPrgErr	Menu Manager error
32767	dsSysErr	General system error

Notes: * Assembly-language constants.

Source: Inside Macintosh, pages II-364 to II-366, IV-231, V-572

See Also: 3.252. System Error Handler Routine

3.254 Operating System Utility Routines

Name	*Description*	*Defined **	*Pascal Definition*
Date2Secs	Converts date/time into seconds since 1/1/04	II	PROCEDURE Date2Secs (date:DateTimeRec; VAR secs:LONGINT);
Delay	Causes system to wait for ticks	II	PROCEDURE Delay (numTicks:LONGINT; VAR finalTicks:LONGINT);
Dequeue	Removes entry from queue	II	FUNCTION Dequeue (qEntry:QElemPtr; theQueue:QHdrPtr) : OSErr;
Enqueue	Appends entry to queue	II	PROCEDURE Enqueue (qEntry:QElemPtr; theQueue:QHdrPtr);
Environs	Returns current ROM version number	II	PROCEDURE Environs (VAR rom,machine:INTEGER);
EqualString	Returns TRUE if strings are equal	II	FUNCTION EqualString (aStr,bStr:Str255; caseSens,diacSens:BOOLEAN) : BOOLEAN;
GetDateTime	Returns seconds from low memory	II	PROCEDURE GetDateTime (VAR secs:LONGINT);
GetMMUMode	Returns address translation mode in use	V	FUNCTION GetMMUMode (VAR mode:INTEGER);
GetSysPPtr	Returns pointer to low memory copy of parameter RAM	II	FUNCTION GetSysPPtr : SysPPtr;
GetTime	Returns date/time from seconds	II	PROCEDURE GetTime (VAR date:DateTimeRec);
GetTrapAddress	Returns address of trap routine	II	FUNCTION GetTrapAddress (trapNum:INTEGER) : LONGINT;
HandAndHand	Concatenates aHndl information to bHndl	II	FUNCTION HandAndHand (aHndl,bHndl:Handle) : OSErr;
HandToHand	Returns handle to copy of handle	II	FUNCTION HandToHand (VAR theHandl:Handle) : OSErr;
InitUtil	Copies parameter RAM into low memory	II	FUNCTION InitUtil : OSErr;
NGetTrapAddress	GetTrapAddress with operating system or ToolBox choice	IV	FUNCTION NGetTrapAddress (trapNum:INTEGER; tType:TrapType) : LONGINT;
NSetTrapAddress	SetTrapAddress with operating system or ToolBox choice	IV	PROCEDURE NSetTrapAddress (trapAddr:LONGINT; trapNum:INTEGER; tType:TrapType);
PtrAndHand	Concatenates bytes at Ptr to Handle	II	FUNCTION PtrAndHand (pntr:Ptr; hndl:Handle; size:LONGINT) : OSErr;
PtrToHand	Returns handle to copy of bytes at pointer	II	FUNCTION PtrToHand (srcPtr:Ptr; VAR dstHndl:Handle; size:LONGINT) : OSErr;
PtrToXHand	Makes existing handle a handle to copy of bytes	II	FUNCTION PtrToXHand (srcPtr:Ptr; dstHndl:Handle; size:LONGINT) : OSErr;
ReadDateTime	Copies date/time from clock to memory	II	FUNCTION ReadDateTime (VAR secs:LONGINT) : OSErr;
RelString	Returns <, =, or > for string (-1,0,1)	IV	FUNCTION RelString (aStr,bStr:Str255; caseSens,diacSens:BOOLEAN) : INTEGER;
Restart	Restarts the system	II	PROCEDURE Restart;
RestoreA5	Restores value of A5 register (from SetUpA5)	II	PROCEDURE RestoreA5;
Secs2Date	Converts seconds into date/time	II	PROCEDURE Secs2Date (secs:LONGINT; VAR date:DateTimeRec);
SetDateTime	Writes seconds to clock chip	II	FUNCTION SetDateTime (secs:LONGINT) : OSErr;
SetTime	Converts date/time into seconds, writes to clock	II	PROCEDURE SetTime (date:DateTimeRec);
SetTrapAddress	Installs new trap in dispatch table	II	PROCEDURE SetTrapAddress (trapAddr:LONGINT; trapNum:INTEGER);
SetUpA5	Saves current value of A5 register	II	PROCEDURE SetUpA5;
StripAddress	Returns 24-bit address if in 24-bit mode	V	FUNCTION StripAddress (theAddress:LONGINT) : LONGINT;
SwapMMUMode	Sets address translation mode	V	PROCEDURE SwapMMUMode (VAR mode:Byte);
SysBeep	Causes system to beep for ticks	II	PROCEDURE SysBeep (duration:INTEGER);
UprString	Converts lowercase characters in string to uppercase	II	PROCEDURE UprString (VAR theString:Str255; diacSens:BOOLEAN);
WriteParam	Writes low memory copy of parameter RAM to clock chip	II	FUNCTION WriteParam : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-374 to II-386, IV-234 to IV-236, V-592 to V-593

See Also: 3.156. Operating System Calls
3.255. Queue Types
3.256. DateTimeRec Data Structure
3.257. Operating System Result Codes

3.255. Queue Types

Size	*Name*	*Comments*
INTEGER	qFlags	Queue flags
QElemPtr *	qHead	First queue entry
QElemPtr *	qTail	Last queue entry

Notes: * QElem structure varies depending on the queue type:

Queue Type	*Queue Structure*	*See Also*
vType	VBLTask	3.300. VBLTask Data Structure
ioQType	ParamBlockRec	3.191. ParamBlockRec Data Structure
drvQType	DrvQE1	3.199. DrvQE1 Data Structure
evType	EvQE1	3.174. EvQE1 Data Structure
fsQType	VCB	3.197. VCB Data Structure

Source: Inside Macintosh, pages II-372 to II-373

See Also: 3.254. Operating System Utility Routines

3.256. DateTimeRec Data Structure

Size	*Name*	*Comments*
INTEGER	year	1904 to 2040
INTEGER	month	1 to 12 (January to December)
INTEGER	day	1 to 31
INTEGER	hour	0 to 23 (midnight to 11 p.m.)
INTEGER	minute	0 to 59
INTEGER	second	0 to 59
INTEGER	dayOfWeek	1 to 7 (Sunday to Saturday)

Source: Inside Macintosh, page II-378

See Also: 3.254. Operating System Utility Routines

3.257. Operating System Result Codes

Value	*Name*	*Comments*
-85	clkRdErr	Couldn't read clock
-86	clkWrErr	Time didn't verify after write
-108	memFullErr	Heap zone full
-111	memWZErr	Attempted to operate on a free memory block
-109	nilHandleErr	NIL master pointer
0	noErr	No error
-88	prInitErr	Incorrect validity status (not $A8)
-87	prWrErr	Parameter RAM did not verify after write
-1	qErr	Entry not in queue

Source: Inside Macintosh, page II-387

See Also: 3.254. Operating System Utility Routines

3.258. Printing Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
PrClose	Releases memory used by Printing Manager	II	PROCEDURE PrClose;
PrCloseDoc	Closes printing grafPort	II	PROCEDURE PrCloseDoc (pPrPort:TPPrPort);
PrClosePage	Finishes printing of current page	II	PROCEDURE PrClosePage (pPrPort:TPPrPort);
PrCtlCall	Low-level printer device control routine	II	PROCEDURE PrCtlCall (iWhichCtl:INTEGER; lParam1,lParam2,lParam3:LONGINT);
PrDrvDCE	Low-level routine returns handle to device control entry	II	FUNCTION PrDrvrDCE : Handle;
PrDrvrClose	Low-level printer driver close routine	II	PROCEDURE PrDrvrClose;
PrDrvrOpen	Low-level printer driver open routine	II	PROCEDURE PrDrvrOpen;
PrDrvrVers	Low-level routine returns printer driver version	II	FUNCTION PrDrvrVers : INTEGER;
PrError	Returns result code of last Printing Manager call	II	FUNCTION PrError : INTEGER;
PrintDefault	Fills fields of print record with default values	II	PROCEDURE PrintDefault (hPrint:THPrint);
PrJobDialog	Displays print job dialog	II	FUNCTION PrJobDialog (hPrint:THPrint) : BOOLEAN;
PrJobMerge	Calls PrValidate for print records	II	PROCEDURE PRJobMerge (hPrintSrc,hPrintDst:THPrint);
PrOpen	Opens Printing Manager	II	PROCEDURE PrOpen;
PrOpenDoc	Returns pointer to initialized printing grafPort	II	FUNCTION PrOpenDoc (hPrint:THPrint; pPrPort:TPPrPort; pIOBuf:Ptr) : TPPrPort;
PrOpenPage	Begins new page	II	PROCEDURE PrOpenPage (pPrPort:TPPrPort; pPageFrame:TPRect);
PrPicFile	Prints spooled document	II	PROCEDURE PrPicFile (hPrint:THPrint; pPrPort:TPPrPort; pIOBuf:Ptr; pDevBuf:Ptr; VAR prStatus:TPrStatus);
PrSetError	Stores value in Printing Manager result code variable	II	PROCEDURE PrSetError (iErr:INTEGER);
PrStlDialog	Displays page setup dialog	II	FUNCTION PrStlDialog (hPrint:THPrint) : BOOLEAN;
PrValidate	Checks print record for compatibility with Printing Manager	II	FUNCTION PrValidate (hPrint:THPrint) : BOOLEAN;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-157 to II-165

See Also:
3.156. Operating System Calls
3.259. TPrint Data Structure
3.260. TPrInfo Data Structure
3.261. TPrJob Data Structure
3.262. TPrStatus Data Structure
3.263. Printing Manager Result Codes
3.264. Printing Control Constants
3.265. TGnlData Data Structure
3.266. TRslRg Data Structure
3.267. TRslRec Data Structure
3.268. TSetRslBlk Data Structure
3.269. TDftBitsBlk Data Structure
3.270. TGetRotnBlk Data Structure

3.259. TPrint Data Structure

Size	*Name*	*Comments*
INTEGER	iPrVersion	Printing Manager version
TPRInfo	prInfo	See 3.260.
Rect	rPaper	Rectangle defining paper size
TPrStl1	prSt1	See 3.262.
TPrInfo	prInfoPT	USED INTERNALLY
TPrXInfo	prXInfo	Additional device information
TPrJob	prJob	See 3.261.
38 bytes	printX	NOT USED

Source: Inside Macintosh, page II-149

See Also: 3.258. Printing Manager Routines
3.260. TPrInfo Data Structure
3.261. TPrJob Data Structure
3.262. TPrStatus Data Structure

3.260. TPrInfo Data Structure

Size	*Name*	*Comments*
INTEGER	iDev	USED INTERNALLY
INTEGER	iVRes	Vertical resolution of printer, in dpi
INTEGER	iHRes	Horizontal resolution of printer, in dpi
Rect	rPage	Rectangle defining page

Source: Inside Macintosh, page II-150

See Also: 3.258. Printing Manager Routines
3.259. TPrint Data Structure

3.261. TPrJob Data Structure

Size	*Name*	*Comments*
INTEGER	iFstPage	First page to print
INTEGER	iLstPage	Last page to print
INTEGER	iCopies	Number of copies to print
Signed byte	bJDocLoop	Printing method
BOOLEAN	fFromUsr	USED INTERNALLY
ProcPtr	pIdleProc	Background procedure
StringPtr	pFileName	Spool filename, if any
INTEGER	iFileVol	Spool file volume refNum, if any
Signed byte	bFileVers	Spool file version number, if any
Signed byte	bJobX	USED INTERNALLY

Source: Inside Macintosh, page II-151

See Also: 3.258. Printing Manager Routines
3.259. TPrint Data Structure

3.262. TPrStatus Data Structure

Size	*Name*	*Comments*
INTEGER	iTotPages	Total number of pages in spool file
INTEGER	iCurPage	Current page being printed
INTEGER	iTotCopies	Total number of copies requested
INTEGER	iCurCopy	Current copy being printed
INTEGER	iTotBands	USED INTERNALLY
INTEGER	iCurBand	USED INTERNALLY
BOOLEAN	fPgDirty	TRUE = started printing page
BOOLEAN	fImaging	USED INTERNALLY
THPrint	hPrint	Print record
TPPrPort	pPrPort	Printing grafPort
PicHandle	hPic	USED INTERNALLY

Source: Inside Macintosh, page II-161

See Also: 3.258. Printing Manager Routines

3.263. Printing Manager Result Codes

Value	*Constant*	*Comments*
-108	iMemFullErr	Heap zone full
-27	iIOAbort	I/O error
-17	controlErr	Control instruction not implemented
-1	iPrSavPFil	Saving print file
0	noErr	No error
128	iPrAbort	User or application requested abort

Source: Inside Macintosh, page II-161

See Also: 3.258. Printing Manager Routines

3.264. Printing Control Constants

Value	*Constant*	*Comments*
$00000000	IScreenBits	Screen bitmap
$00000001	IPaintBits	Paint bitmap, square pixels
$00000002	IHiScreenBits	Screen bitmap
$00000003	IHiPaintBits	Paint bitmap, square pixels
$00010000*	IPrReset	Reset printer
$0001FFFD	IPrEvtTop	Top window only
$00020000**	IPrPageEnd	End page
$0002FFFD	IPrEvtAll	Entire screen
$00030000	IPrLineFeed	Perform carriage return only
$0003FFFF	IPrLFSixth	Perform 1/6-inch line feed
$0003FFFF	IPrLFStd	Standard paper advance
$00040000	IPrPageOpen	Open page
$00050000	IPrDocClose	Close document
4	iPrBitsCtl	Bitmap print proc's control number
5	iPrIOCtl	Raw byte I/O proc's control number
6	iPrEvtCtl	PrEvent proc's control number
7	iPrDevCtl	PrDevCtl proc's control number

Notes:
- Used in PrCtlCall parameters.
- * IPrDocOpen alias for IPrReset.
- ** IPrPageClose alias for IPrPageEnd.

Source: Inside Macintosh, page II-163, V-409 to V-410

See Also: 3.258. Printing Manager Routines

3.265. TGnlData Data Structure

Size	*Name*	*Comments*
INTEGER	iOpCode	Routine selector *
INTEGER	iError	Error (result code)
LONGINT	lReserved	RESERVED
Varies	Varies	See 3.266. to 3.270.

Notes: * One of the following codes:

Value	*Operation*
4	GetRslData
5	SetRsl
6	DraftBits
7	NoDraftBits
8	GetRotn

Source: Inside Macintosh, pages V-410 to V-416

See Also:
3.258. Printing Manager Routines
3.266. TRslRg Data Structure
3.267. TRslRec Data Structure
3.268. TSetRslBlk Data Structure
3.269. TDftBitsBlk Data Structure
3.270. TGetRotnBlk Data Structure

3.266. TRslRg Data Structure

Size	*Name*	*Comments*
INTEGER	iMin	Minimum resolution
INTEGER	iMax	Maximum resolution

Notes: Appended to end of TGnlData data structure if iOpCode = 4.

Source: Inside Macintosh, pages V-411 to V-412

See Also: 3.258. Printing Manager Routines
3.265. TGnlData Data Structure

3.267. TRslRec Data Structure

Size	*Name*	*Comments*
INTEGER	iXRsl	Physical X resolution
INTEGER	iYRsl	Physical Y resolution

Notes: Appended to end of TGnlData data structure if iOpCode = 4.

Source: Inside Macintosh, pages V-411 to V-412

See Also: 3.258. Printing Manager Routines
3.265. TGnlData Data Structure

3.268. TSetRslBlk Data Structure

Size	*Name*	*Comments*
INTEGER	iOpCode	= 5, setRslOp
INTEGER	iError	Result code output
LONGINT	lReserved	RESERVED
THPrint	hPrint	Handle to valid print record
INTEGER	iXRsl	Desired X resolution
INTEGER	iYRsl	Desired Y resolution

Source: Inside Macintosh, page V-414

See Also: 3.258. Printing Manager Routines
3.265. TGnlData Data Structure

3.269. TDftBitsBlk Data Structure

Size	*Name*	*Comments*
INTEGER	iOpCode	= 6, DraftBits
INTEGER	iError	Result code output
LONGINT	lReserved	RESERVED
THPrint	hPrint	Handle to valid print record

Source: Inside Macintosh, pages V-414 to V-415

See Also: 3.258. Printing Manager Routines
3.265. TGnlData Data Structure

3.270. TGetRotnBlk Data Structure

Size	*Name*	*Comments*
INTEGER	iOpCode	= 8, getRotnOp
INTEGER	iError	Result code output
LONGINT	lReserved	RESERVED
THPrint	hPrint	Handle to valid print record
BOOLEAN	fLandscape	Flag (TRUE if landscape mode)
Signed byte	bXtra	RESERVED

Source: Inside Macintosh, page V-415

See Also: 3.258. Printing Manager Routines
3.265. TGnlData Data Structure

3.271. Slot Manager Routines

				spBlockPtr Use		
Name	*Description*	*Defined **	*Pascal Definition*	*Input*	*Output*	*Affected*
SRsrcInfo	Returns sResource list pointer and other slot card information	V	FUNCTION SRsrcInfo (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spId spExtDev	spsPointer (sResource list) spIOReserved spRefNum spCategory spCType spDrvrSW spDrvrHW spHWDev	
SNextsRsrc	Returns resource information from sResource	V	FUNCTION SNextsRsrc (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID spExtDev	spSlot spID spExtDev spsPointer spRefNum spIOReserved spCategory spCType spDrvrSW spDrvrHW spHWDev	
SNextTypesRsrc	Like SNextsRsrc, but only for specific resource type	V	FUNCTION SNextTypesRsrc (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID spExtDev spTBMask spCategory spCType spDrvrSW spDrvrHW spHWDev	spSlot spID spExtDev spsPointer spRefNum spIOReserved spCategory spCType spDrvrSW spDrvrHW spHWDev	
SReadDrvrName	Returns name of driver from slot resource	V	FUNCTION SReadDrvrName (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID spResult	 spResult (points at name)	spSize spsPointer
SReadByte	Returns byte from slot card firmware	V	FUNCTION SReadByte (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spID	spResult	spOffsetData spByteLanes
SReadWord	Returns word from slot card firmware	V	FUNCTION SReadWrod (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spID	spResult	spOffsetData spByteLanes
SReadLong	Returns LONGINT from slot card firmware	V	FUNCTION SReadLong (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spID	spResult	spOffsetData spByteLanes spSize
SGetCString	Copies string from slot card to buffer pointed to by spResult	V	FUNCTION SGetCString (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spID	spResult	spOffsetData spByteLanes spSize spFlags

(continued)

Table 3.271. Continued

				spBlockPtr Use		
Name	*Description*	*Defined **	*Pascal Definition*	*Input*	*Output*	*Affected*
SGetBlock	Copies block from slot card to block pointed to by spResult	V	FUNCTION SGetBlock (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer splD	spResult	spOffsetData spByteLanes spSize spFlags
SFindStruct	Returns pointer to data structure in slot card	V	FUNCTION SFindStructure (spBlkPtr:SpBlockPtr) : OSErr;	splD spsPointer	spsPointer	spByteLanes
SReadStruct	Copies structure from slot card to spsPointer location	V	FUNCTION SReadStruct (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spSize spResult		spByteLanes
SReadInfo	Reads slot card record to new record at spResult	V	FUNCTION SReadInfo (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spResult	copy of spSlot at spResult	spSize
SReadPRAMRec	Copies sPRAM data to block pointed to by spResult	V	FUNCTION SReadPRAMRec (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spResult	spResult (sPRAM copy)	spSize
SPutPRAMRec	Copies block at spsPointer to sPRAM record	V	FUNCTION SPutPRAMRec (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spPointer		
SReadFHeader	Copies format block data to FHeader at spResult	V	FUNCTION SReadFHeader (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spResult	spResult (FHeader copy)	spsPointer spByteLanes spSize spOffsetData
SCkCardStatus	Returns InitStatus A field of sInfo into spResult	V	FUNCTION SCkCardStatus (spBlkPtr:spBlockPtr) : OSErr;	spSlot	spResult	spResult
SFindDevBase	Returns pointer in spResult to device base	V	FUNCTION SFindDevBase (spBlkPtr:spBlockPtr) : OSErr;	spSlot splD	spResult (pointer to base)	
SDeleteSRTRec	Deletes sResource from slot resource table	V	FUNCTION SDeleteSRTRec (spBlkPtr:spBlockPtr) : OSErr;	spSlot splD spExtDev		
SPtrToSlot	Returns slot number of card at spsPointer	V	FUNCTION SPtrToSlot (spBlkPtr:spBlockPtr) : OSErr;	spsPointer	spSlot	
InitSDeclMgr	Initializes Slot Manager	V	FUNCTION InitSDeclMgr (spBlkPtr:SpBlockPtr) : OSErr;	undefined	undefined	Undefined
SPrimaryInit	Initializes all slots with SPrimary Init record	V	FUNCTION SPrimaryInit (spBlkPtr:SpBlockPtr) : OSErr;	spFlags		
SCardChanged	Returns TRUE in spResult if card has been changed	V	FUNCTION SCardChanged (spBlkPtr:SpBlockPtr) : OSErr;	spSlot	spResult	
SExec	Loads SExec code block at spsPointer and executes it	V	FUNCTION SExec (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer splD spsExecPBlk		spResult
SOffsetData	Returns contents of offset/data field	V	FUNCTION SOffsetData (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer splD	spOffsetData spByteLanes	spResult spFlags
SReadPBSize	Returns physical block size of sBlock at spsPointer	V	FUNCTION SReadPBSize (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer splD spFlags	spSize spByteLanes	spResult
SCalcStep	Calculates field sizes in block	V	FUNCTION SCalcStep (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spByteLanes spFlags	spResult	
InitsRsrcTable	Initializes slot resource table	V	FUNCTION InitsRsrcTable (spBlkPtr:SpBlockPtr) : OSErr;	Undefined	Undefined	Undefined
InitPRAMRecs	Initializes PRAM records for each card	V	FUNCTION InitPRAMRecs (spBlkPtr:SpBlockPtr) : OSErr;	Undefined	Undefined	Undefined

(continued)

Table 3.271. Continued

				spBlockPtr Use		
Name	*Description*	*Defined **	*Pascal Definition*	*Input*	*Output*	*Affected*
SSearchSRT	Searches slot resource table for record, returns pointer in spsPointer	V	FUNCTION SSearchSRT (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID spExtDev spFlags spsPointer	spsPointer (ptr to record)	
SUpdateSRT	Updates slot resource table	V	FUNCTION SUpdateSRT (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID spExtDev spRefNum spIOReserved		spsPointer spFlags spSize spResult
SCalcSPtr	Returns pointer to card's declaration ROM	V	FUNCTION SCalcSPtr (spBlkPtr:SpBlockPtr) : OSErr;	spsPointer spOffsetData spByteLanes		
SGetDriver	Returns handle to loaded driver in spResult	V	FUNCTION SGetDriver (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID spExtDev spsExecPBlk	spResult	spFlags spSize
SFindsInfoRecPtr	Returns pointer to sInfoRecord	V	FUNCTION SFindsInfoRecPtr (spBlkPtr:SpBlockPtr) : OSErr;	spSlot	spResult	
SFindsRsrcPtr	Returns pointer to sRsrc list	V	FUNCTION SFindsRsrcPtr (spBlkPtr:SpBlockPtr) : OSErr;	spSlot spID	spsPointer	spResult

Notes: • These calls all require you to set parameters in SpBlkPtr and return values in same.
•* Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-437 to V-456

See Also: 3.156. Operating System Calls
3.272. SExecBlock Data Structure
3.273. SpBlock Data Structure
3.274. SInfoRecord Data Structure
3.275. SResource Directory Data Structure
3.276. SResource Apple-Defined IDs
3.277. SPRAMRecord Data Structure
3.278. FHeaderRec Data Structure
3.279. VendorInfo List IDs
3.280. Slot Manager Result Codes

3.272. SExecBlock Data Structure

Size	*Name*	*Comments*
Signed byte	seSlot	Slot number
Signed byte	sesRsrcId	ID for sResource
INTEGER	seStatus	Status of sExec code
Signed byte	seFlags	Flags
Signed byte	seFiller0	Filler to align word boundary
Signed byte	seFiller1	FILLER
Signed byte	seFiller2	FILLER
LONGINT	seResult	sLoad result
LONGINT	selOFileName	Pointer to IOFile name
Signed byte	seDevice	Device to read from
Signed byte	sePartition	Partition
Signed byte	seOSType	Type of operating system
Signed byte	seReserved	RESERVED
Signed byte	seRefNum	Driver refNum
Signed byte	seNumDevices	Number of devices to load
Signed byte	seBootState	StartBoot code status

Source: Inside Macintosh, page V-441

See Also: 3.271. Slot Manager Routines

3.273. SpBlock Data Structure

Size	*Name*	*Comments*
LONGINT	spResult	Result code from function
Ptr	spsPointer	Structure pointer
LONGINT	spSize	Structure size, in bytes
LONGINT	spOffsetData	Offset/data field
Ptr	splOFileName	Pointer to IOFile name
Ptr	spsExecPBlk	Pointer to sExec param block
Ptr	spStackPtr	Old stack pointer
LONGINT	spMisc	Miscellaneous field
LONGINT	spReserved	RESERVED
INTEGER	splOReserved	Reserved field of slot resource table
INTEGER	spRefNum	RefNum
INTEGER	spCategory	sType category
INTEGER	spCType	sType type
INTEGER	spDrvrSW	sType DrvrSW
INTEGER	spDrvrHW	sType DrvrHW
Signed byte	spTBMask	Type bitmask *
Signed byte	spSlot	Slot number
Signed byte	splD	Structure ID
Signed byte	spExtDev	External device ID
Signed byte	spHWDev	Hardware device ID
Signed byte	spByteLanes	ByteLanes from card ROM
Signed byte	spFlags	Standard flags
Signed byte	spKey	USED INTERNALLY

Notes: * Bits 0–3.

Source: Inside Macintosh, pages V-439 to V-440

See Also: 3.271. Slot Manager Routines

3.274. SInfoRecord Data Structure

Size	*Name*	*Comments*
Ptr	siDirPtr	Pointer to directory
INTEGER	siInitStatusA	Initialization error, if any
INTEGER	siInitStatusV	Status returned by vendor init
Signed byte	siState	Initialization state
Signed byte	siCPUByteLanes	*
Signed byte	siTopOfROM	Top of ROM **
Signed byte	siStatusFlags	Bit zero represents card changed status ***
INTEGER	siTOConstant	Timeout constant for bus error
Signed byte	siReserved	RESERVED

Notes:
- * 0 = d0 to d7, 1 = d8 to d15.
- ** Normally $FsFFFFFx, where s is slot number, x is top of ROM.
- *** 1 = card changed, 0 = still same card.

Source: Inside Macintosh, page V-447

See Also: 3.271. Slot Manager Routines

3.275. SResource Directory Data Structure

Size	*Name*	*Comments*
LONGINT	sRsrcID0 *	Offset to first directory entry
LONGINT	sRsrcID1 *	
LONGINT	sRsrcIDn *	Offset to last directory entry
LONGINT	endOfList	Should be $FF00 0000

Notes:
- * First 8 bits are ID, LO 24 bits are offset to sResource list.

Each sResource list is formatted as follows:

Size	*Name*	*Comments*
LONGINT	sRsrcType	8 bit ID, 24 bit offset
LONGINT	sRsrcName	8 bit ID, 24 bit offset
LONGINT	sRsrcIcon	Same as above or **
LONGINT	endOfList	Should be $FF00 0000

- ** HO byte = ID, MO byte = $00, LO word = word data.

HO byte = ID, MO byte = $00 00, LO byte = byte data.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 8-9 to 8-10

See Also: 3.271. Slot Manager Routines

3.276. SResource Apple-Defined IDs

Value	*Constant*	*Comments*
1	sRsrc_Type	sResource type (required)
2	sRsrc_Name	sResource name (required)
3	sRsrc_Icon	sResource icon
4	sRsrc_DrvrDir	sResource driver directory
5	sRsrc_LoadRec	sResource load record
6	sRsrc_BootRec	Boot record
7	sRsrc_Flags	sResource flags
8	sRsrc_HWDevId	Hardware device ID
10	MinorBaseOS	Offset to base of sResource in slot space
11	MinorLength	Length of sResource, in bytes
12	MajorBaseOS	Offset to base of sResource in super slot space
13	MajorLength	Length of sResource, in bytes

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 8-11

See Also: 3.271. Slot Manager Routines
3.275. sResource Directory Data Structure

3.277. SPRAMRecord Data Structure

Size	*Name*	*Comments*
INTEGER	boardID	Apple-designated card ID
Byte	vendorUse1	Vendor-defined
Byte	vendorUse2	Vendor-defined
Byte	vendorUse3	Vendor-defined
Byte	vendorUse 4	Vendor-defined
Byte	vendorUse5	Vendor-defined
Byte	vendorUse6	Vendor-defined

Source: Inside Macintosh, page V-448

See Also: 3.271. Slot Manager Routines

3.278. FHeaderRec Data Structure

Size	*Name*	*Comments*
LONGINT	fhDIROffset	Offset to directory
LONGINT	fhLength	ROM length, in bytes
LONGINT	fhCRC	CRC value for ROM
Signed byte	fhROMRev	ROM revision number
Signed byte	fhFormat	Format - 2
LONGINT	fhTstPat	Test pattern
INTEGER	fhReserved	RESERVED
Signed byte	fhByteLanes	ByteLanes value

Source: Inside Macintosh, page V-449

See Also: 3.271. Slot Manager Routines

3.279. VendorInfo List IDs

Value	*Constant*
1	VendorID
2	SerialNum
3	RevLevel
4	PartNum
5	Date

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 8-19

See Also: 3.271. Slot Manager Routines
3.275. sResource Directory Data Structure

3.280. Slot Manager Result Codes

Value	*Fatal?*	*Constant*	*Comments*
-351	N	smRecNotFnd	Record not found in slot resource table
-350	N	smSRTOvrFlErr	Slot resource table overflow
-347	N	smByteLanesErr	Bad ByteLanes value
-346	N	smBadsPtrErr	Bad sPointer
-345	N	smGetDrvrErr	Error during sGetDrvr
-344	N	smNoMoresRsrcs	No more sResources
-342	N	smGetDrvrNamErr	Error during sGetDrvrName
-341	N	smCkStatus	Slot status is bad
-339	N	smNewPErr	Error during NewPointer
-338	N	smSelOOBErr	Selector is out of bounds
-337	N	smSlotOOBErr	Slot is out of bounds or does not exist
-336	N	smNilsBlockErr	Physical block size was zero
-335	N	smsPointerNil	SPointer is NIL
-334	N	smCPUErr	CPU field of sExec was wrong
-333	N	smCodeRevErr	Revision of sExec code was wrong
-332	N	smReservedErr	Reserved field was not zero
-331	N	smBadsList	IDs in sResource not in ascending order
-330	N	smBadRefID	No ref ID in sResource
-319	Y	smBadBoardID	BoardID wrong
-318	Y	smNoJmpTbl	Slot manager jump table not created
-317	Y	smInitTblErr	Error during init of sResource table
-316	Y	smInitStatVErr	InitStatusV field negative after primary init
-315	Y	smNoBoardID	No board ID
-314	Y	smGetPRErr	Error during sGetPRAMRec
-313	Y	smNoBoardsRsrc	No board sResource
-312	Y	smDisposePerr	Error during DisposPointer
-309	Y	smBLFieldBad	No valid ByteLanes found
-308	Y	smUnExBusErr	Unexpected bus error
-307	Y	smReservErr	Reserved field of declaration ROM was used
-306	Y	smNosInfoArray	SDM unable to allocate memory for array
-304	Y	smNoDir	No sResource directory
-303	Y	smRevisionErr	Declaration ROM revision invalid
-302	Y	smFormatErr	Declaration ROM format invalid
-301	Y	smCRCFail	CRC failed
-300	Y	smEmptySlot	No card in slot

Source: Inside Macintosh, pages V-456 to V-457

See Also: 3.271. Slot Manager Routines

3.281. Start Manager Routines

				paramBlock Use	
Name	*Description*	*Defined **	*Pascal Definition*	*Input*	*Output*
GetDefaultStartup	Returns default startup information from PRAM	V	PROCEDURE GetDefaultStartup (paramBlock:DefStartPtr);		sdExtDevID or dsReserved1 sdPartition or sdReserved2 sdSlotNum or sdRefNum sdSRsrcID
SetDefaultStartup	Changes default startup information	V	PROCEDURE SetDefaultStartup (paramBlock:DefStartPtr);	sdExtDevID or sdReserved1 sdPartition or sdReserved2 sdSlotNum or sdRefNum sdSRsrcID	
GetVideoDefault	Returns slot number and sResource ID of video	V	PROCEDURE GetVideoDefault (paramBlock:DefVideoPtr);		sdSlot sdSResource
SetVideoDefault	Changes default video device	V	PROCEDURE SetVideoDefault (paramBlock:DefVideoPtr);	sdSlot sdSResource	
GetOSDefault	Returns startup operating system type	V	PROCEDURE GetOSDefault (paramBlock:DefOSPtr);		sdReserved sdOSType
SetOSDefault	Specifies default operating system	V	PROCEDURE SetOSDefault (paramBlock:DefOSPtr);	sdReserved sdOSType	
GetTimeOut	Returns number of seconds system waits for HD	V	PROCEDURE GetTimeOut (VAR count:INTEGER);		d0 = count
SetTimeOut	Specifies HD wait at startup	V	PROCEDURE SetTimeOut (Count:INTEGER);	d0 = count	

Notes: • These calls all require you to set parameters in paramBlock and return values in same.
•* Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-352 to V-356

See Also: 3.156. Operating System Calls
3.282. DefStartRec Data Structure
3.283. DefVideoRec Data Structure
3.284. DefOSRec Data Structure

3.282. DefStartRec Data Structure

For Slot Devices

Size	*Name*	*Comments*
Signed byte	sdExtDevID	External device ID
Signed byte	sdPartition	RESERVED
Signed byte	sdSlotNum	Slot number
Signed byte	sdSRsrcID	SResource ID

For SCSI Devices

Signed byte	sdReserved1	RESERVED
Signed byte	sdReserved2	RESERVED
INTEGER	sdRefNum	Driver refNum

Source: Inside Macintosh, page V-353

See Also: 3.281. Start Manager Routines

3.283. DefVideoRec Data Structure

Size	*Name*	*Comments*
Signed byte	sdSlot	Slot number
Signed byte	sdSResource	sResourceID

Source: Inside Macintosh, page V-357

See Also: 3.281. Start Manager Routines

3.284. DefOSRec Data Structure

Size	*Name*	*Comments*
Signed byte	sdReserved	RESERVED *
Signed byte	sdOSType	Operating system type

Note: * Should be 0.

Source: Inside Macintosh, page V-357

See Also: 3.281. Start Manager Routines

3.285. Shutdown Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
ShutDwnPower	Housekeeps, executes shutdown procedures, turns off machine	V	PROCEDURE ShutDwnPower;
ShutDwnStart	Housekeeps, executes shutdown procedures, reboots	V	PROCEDURE ShutDwnStart;
ShutDwnInstall	Installs shutdown procedure	V	PROCEDURE ShutDwnInstall (shutDwnProc:ProcPtr; flags:INTEGER);
ShutDwnRemove	Removes shutdown procedure	V	PROCEDURE ShutDwnRemove (shutDwnProc:ProcPtr);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-587 to V-588

See Also: 3.156. Operating System Calls
3.286. ShutDwnInstall Masks

3.286. ShutDwnInstall Masks

Value	*Constant*	*Comments*
1	sdOnPowerOff	Call the procedure before power off
2	sdOnRestart	Call the procedure before restart
4	sdOnUnmount	Call the procedure before unmounting
8	sdOnDrivers	Call the procedure before closing drivers

Notes: Tells the Shutdown Manager when the procedure is to be executed.

Source: Inside Macintosh, page V-589

See Also: 3.285. Shutdown Manager Routines

3.287. ADB Bus Routines

				A0/D0/Parameter Block Use			
Name	*Description*	*Defined **	*Pascal Definition*	*Input*	*ParamBlock In*	*Output*	*ParamBlock Out*
ADBReInit	Reinitializes ADB Bus	V	PROCEDURE ADBReInit;				
ADBOp	Transmits command byte over bus	V	FUNCTION ADBOp (data:Ptr; compRout:ProcPtr; buffer:Ptr; commandNum:INTEGER) : OSErr;	D0 = commandNum byte A0 = ptr to param block	buffer (ptr) compRout (ptr) data (ptr)	D0 = result code byte	
CountADBs	Returns count of connected ADB devices	V	FUNCTION CountADBs : INTEGER;			D0 = number of devices byte	
GetIndADB	Returns info from ADB device entry table	V	FUNCTION GetIndADB (VAR info:ADBDataBlock; devTableIndex:INTEGER) : ADBAddress;	D0 = entry index number A0 = ptr to param block		D0 = positive: current ADB addr D0 = negative: error code	device type (byte) orig ADB addr (byte) service routine addr (ptr) data area addr (ptr)
GetADBInfo	Returns info from ADB device entry table	V	FUNCTION GetADBInfo (VAR info:ADBDataBlock; ADBAddr:ADBAddress) : OSErr;	D0 = ADB address byte A0 = ptr to param block		D0 = result code	device handler ID (byte) orig ADB addr (byte) service routine addr (ptr) data area addr (ptr)
SetADBInfo	Changes ADB device entry table	V	FUNCTION SetADBInfo (VAR info:ADBSetInfoBlock; ADBAddr:ADBAddress) : OSErr;	D0 = ADB address byte A0 = ptr to param block	service routine addr (ptr) data area addr (ptr)	D0 = result code	

Notes: • These calls all require you to set parameters in paramBlock and return values in same.
•* Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-367 to V-371

See Also: 3.156. Operating System Calls
3.288. ADB Bus Command Formats
3.289. ADB Bus Register 3 Data Fields
3.290. ADBDataBlock Data Structure
3.291. ADBSetInfoBlock Data Structure

3.288. ADB Bus Command Formats

SendReset Command:

Bit Numbers

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Description*
X	X	X	X					NOT USED
				0	0	0	0	SendReset command

Flush Command:

X	X	X	X					Network address
				0	0	0	1	Flush command

Listen Command:

X	X	X	X					Network address
				1	0			Listen command
						X	X	Register number

Talk Command:

X	X	X	X					Network address
				1	1			Talk command
						X	X	Register number

Source: Inside Macintosh, page V-363

See Also: 3.287. ADB Bus Routines

3.289. ADB Bus Register 3 Data Fields

Bit Numbers

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Description*
0																Must be zero
	X															Exceptional event flag
		X														Service request enable flag
			X													RESERVED
				X	X	X	X									Device address
								X	X	X	X	X	X	X	X	Device handler ID

Source: Inside Macintosh, page V-365

See Also: 3.287. ADB Bus Routines

3.290. ADBDataBlock Data Structure

Size	*Name*	*Comments*
Signed byte	devType	Type of device
Signed byte	origADBAddr	Original ADB address
Ptr	dbServiceRtPtr	Service routine address
Ptr	dbDataAreaAddr	Data area address

Source: Inside Macintosh, page V-369

See Also: 3.287. ADB Bus Routines

3.291. ADBSetInfoBlock Data Structure

Size	*Name*	*Comments*
Ptr	siServiceRtPtr	Service routine address
Ptr	siDataAreaAddr	Data area address

Source: Inside Macintosh, page V-370

See Also: 3.287. ADB Bus Routines

3.292. Video Driver Parameter IDs

Value	*Constant*	*Comments*
1	mVidParams	Video device record ID
2	mTable	Offset to table
3	mPageCnt	Number of pages in 1-bit mode
4	mDevType	Device type

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 9-11

See Also: 3.293. VideoDeviceRec Data Structure
3.294. Video Driver Data Structures

3.293. VideoDeviceRec Data Structure

Size	*Name*	*Comments*
Long	vpBaseOffset	Offset from MinorBaseOS to base page (0) of video RAM
Word	vpRowBytes	Width of each row of video memory, in pixels
4 words	vpBounds	BoundsRect for video display
Word	vpVersion	PixelMap version number
Word	vpPackType	UNUSED
Word	vpPackSize	UNUSED
Long	vpHRes	Horizontal resolution of display, in pixels per inch
Long	vpVRes	Vertical resolution of display, in pixels per inch
Word	vpPixelType	Pixel type *
Word	vpPixelSize	Pixel depth, in bits per pixel
Word	vpCmpCount	Number of components in pixel
Word	vpCmpSize	Size of components, in bits
Long	vpPlaneBytes	Offset from one plane to next

Notes: * Chunky = 0, planar = 1.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 9-12

3.294. Video Driver Data Structures

VDEntryRecord

Size	*Name*	*Comments*
Ptr	csTable	Pointer to color lookup table
INTEGER	csStart	Starting entry number
INTEGER	csCount	Count number

VDGammaRecord

Ptr	csGTable	Pointer to gamma table

VDPgInfo

INTEGER	csMode	Device mode
LONGINT	csData	Driver-supplied data
INTEGER	csPage	Page to switch in
Ptr	csBaseAddr	Base address of page to switch in

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 9-14

3.295. Color Picker Package Routines

Name	*Description*	*Defined **	*Pascal Definition*
GetColor	Displays Color Picker dialog	V	FUNCTION GetColor (where:Point; prompt:Str255; inColor:RGBColor; VAR outColor:RGBColor) : BOOLEAN;
CMY2RGB	Converts CMY color to RGB	V	PROCEDURE CMY2RGB (cColor:CMYColor; VAR rColor:RGBColor);
RGB2CMY	Converts RGB color to CMY	V	PROCEDURE RGB2CMY (rColor:RGBColor; VAR cColor:CMYColor);
HSL2RGB	Converts HSL color to RGB	V	PROCEDURE HSL2RGB (hColor:HSLColor; VAR rColor:RGBColor);
RGB2HSL	Converts RGB color to HSL	V	PROCEDURE RGB2HSL (rColor:RGBColor; VAR hColor:HSLColor);
HSV2RGB	Converts HSV color to RGB	V	PROCEDURE HSV2RGB (hColor:HSVColor; VAR rColor:RGBColor);
RGB2HSV	Converts RGB color to HSV	V	PROCEDURE RGB2HSV (rColor:RGBColor; VAR hColor:HSVColor);
Fix2SmallFract	Converts fixed to fractional	V	FUNCTION Fix2SmallFract (f:Fixed) : SmallFract;
SmallFract2Fix	Converts fractional to fixed	V	FUNCTION SmallFract2Fix (s:SmallFract) : Fixed;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages V-174 to V-175

See Also: 3.035. RGBColor Data Structure
3.156. Operating System Calls
3.296. HSVColor Data Structure
3.297. HSLColor Data Structure
3.298. CMYColor Data Structure

3.296. HSVColor Data Structure

Size	*Name*	*Comments*
SmallFract	hue	Fraction of circle, with red at 0
SmallFract	saturation	0–1: 0 = gray, 1 = pure color
SmallFract	value	0–1: 0 = black, 1 = maximum intensity

Source: Inside Macintosh, page V-176

See Also: 3.035. RGBColor Data Structure
3.295. Color Picker Package Routines
3.297. HSLColor Data Structure
3.298. CMYColor Data Structure

3.297. HSLColor Data Structure

Size	*Name*	*Comments*
SmallFract	hue	Fraction of circle, with red at 0
SmallFract	saturation	0–1: 0 = gray, 1 = pure color
SmallFract	lightness	0–1: 0 = black, 1 = white

Source: Inside Macintosh, page V-176

See Also: 3.035. RGBColor Data Structure
3.295. Color Picker Package Routines
3.296. HSVColor Data Structure
3.298. CMYColor Data Structure

3.298. CMYColor Data Structure

Size	*Name*
SmallFract	cyan
SmallFract	magenta
SmallFract	yellow

Source: Inside Macintosh, page V-176

See Also: 3.035. RGBColor Data Structure
3.295. Color Picker Package Routines
3.296. HSVColor Data Structure
3.297. HSLColor Data Structure

3.299. Vertical Retrace Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
AttachVBL	Makes theSlot the primary video slot	V	FUNCTION AttachVBL (theSlot:INTEGER) : OSErr;
DoVBLTask	Executes VBL tasks for slot in queue	V	FUNCTION DoVBLTask (theSlot:INTEGER) : OSErr;
GetVBLQHdr	Returns pointer to header of vertical retrace queue	II	FUNCTION GetVBLQHdr : QHdrPtr;
SlotVInstall	Adds slot-oriented VBL task to vertical retrace queue	V	FUNCTION SlotVInstall (vblTaskPtr:QElemPtr; theSlot:INTEGER) : OSErr;
SlotVRemove	Removes slot-oriented VBL task from vertical retrace queue	V	FUNCTION SlotVRemove (vblTaskPtr:QElemPtr; theSlot:INTEGER) : OSErr;
VInstall	Adds VBL task to vertical retrace queue	II	FUNCTION VInstall (vblTaskPtr:QElemPtr) : OSErr;
VRemove	Removes VBL task from vertical retrace queue	II	FUNCTION VRemove (vblTaskPtr:QElemPtr) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-351 to II-352, V-567 to V-568

See Also: 3.156. Operating System Calls
3.300. VBLTask Data Structure
3.301. Vertical Retrace Manager Result Codes

3.300. VBLTask Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
ProcPtr	vblAddr	Pointer to VBL task
INTEGER	vblCount	Task frequency
INTEGER	vblPhase	Task phase (usually 0)

Source: Inside Macintosh, page II-350

See Also: 3.299. Vertical Retrace Manager Routines

3.301. Vertical Retrace Manager Result Codes

Value	*Constant*	*Comments*
-2	vTypErr	qType field is wrong type
-1	qErr	Task entry wasn't in queue
0	noErr	No error

Source: Inside Macintosh, page II-353

See Also: 3.299. Vertical Retrace Manager Routines

3.302. AppleTalk Routines

High-Level Pascal-Defined Routines

Name	*Description*	*Defined **	*Pascal Definition*	*Prior to Call*	*On Return*
MPPOpen	Attempts to load .MPP driver, if not already loaded	II	FUNCTION MPPOpen : OSErr;	NOT USED	NOT USED
MPPClose	Removes .MPP driver and data from memory	II	FUNCTION MPPClose : OSErr;	NOT USED	NOT USED
LAPOpenProtocol	Adds ALAP protocol type to protocol table	II	FUNCTION LAPOpenProtocol (theLAPType:ABByte; protoPtr:Ptr) : OSErr;	NOT USED	NOT USED
LAPCloseProtocol	Removes protocol from protocol table	II	FUNCTION LAPCloseProtocol (theLAPType:ABByte) : OSErr;	NOT USED	NOT USED
LAPWrite	Sends frame to another node	II	FUNCTION LAPWrite (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** lapAddress.dstNodeID lapAddress.lapProtType lapReqCount lapDataPtr	abOpcode (always tLAPWrite) abResult
LAPRead	Receives frame from another node	II	FUNCTION LAPRead (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** lapAddress.lapProtType lapReqCount lapDataPtr	abOpcode (always tLAPRead) abResult lapAddress.dstNodeID lapAddress.srcNodeID lapActCount
LAPRdCancel	Dequeues LAPRead call if no packet arrived yet	II	FUNCTION LAPRdCancel (abRecord:ABRecHandle) : OSErr;	Same as LAPRead	check abResult if error
DDPOpenSocket	Adds socket and listener to socket table	II	FUNCTION DDPOpenSocket (VAR theSocket:Byte; sktListener:Ptr) : OSErr;	NOT USED	NOT USED
DDPCloseSocket	Closes socket and cancels pending DDPRead calls	II	FUNCTION DDPCloseSocket (theSocket:Byte) : OSErr;	NOT USED	NOT USED
DDPWrite	Sends datagram to another socket	II	FUNCTION DDPWrite (abRecord:ABRecHandle; doChecksum:BOOLEAN; async:BOOLEAN) : OSErr;	abUserReference ** ddpType ddpSocket ddpAddress ddpReqCount ddpDataPtr	abOpcode (always tDDPWrite) abResult
DDPRead	Receives datagram from another socket	II	FUNCTION DDPRead (abRecord:ABRecHandle; retCksumErrs:BOOLEAN; async:BOOLEAN) : OSErr;	abUserReference ** ddpSocket ddpReqCount ddpDataPtr	abOpcode (always tDDPRead) abResult ddpType ddpAddress ddpActCount ddpNodeID
DDPRdCancel	Dequeues DDPRead call if no datagram arrived	II	FUNCTION DDPRdCancel (abRecord:ABRecHandle) : OSErr;	Same as DDPRead	check abResult if error
ATPLoad ***	Loads .MPP and .ATP, if necessary	II	FUNCTION ATPLoad : OSErr;	NOT USED	NOT USED
ATPUnload ****	Marks .ATP driver as purgeable	II	FUNCTION ATPUnload : OSErr;	NOT USED	NOT USED
ATPOpenSocket	Opens socket for incoming requests	II	FUNCTION ATPOpenSocket (addrRcvd:AddrBlock; VAR atpSocket:Byte) : OSErr;	NOT USED	NOT USED
ATPCloseSocket	Closes socket	II	FUNCTION ATPCloseSocket (atpSocket:Byte) : OSErr;	NOT USED	NOT USED
ATPSndRequest	Sends request to another socket	II	FUNCTION ATPSndRequest (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** atpAddress atpReqCount atpDataPtr atpRspBDSPtr atpUserData atpXO atpTimeOut atpRetries atpNumBufs	abOpcode (always tATPSndRequest) abResult atpEOM atpNumRsp

(continued)

Table 3.302. Continued

Name	*Description*	*Defined **	*Pascal Definition*	*Prior to Call*	*On Return*
ATPRequest	Sends request to another socket but doesn't require BDS structure for response buffers	II	FUNCTION ATPRequest (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** atpAddress atpReqCount atpDataPtr atpUserData atpXO atpTimeOut atpRetries atpRspBuf atpRspSize	abOpcode (always tATPRequest) abResult atpActCount atpEOM atpRspUData
ATPReqCancel	Dequeues ATPSndRequest/ATPRequest	II	FUNCTION ATPReqCancel (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	Same as ATPSndRequest	Check abResult if error
ATPGetRequest	Gets ready to receive ATPSndRequest or ATPRequest call	II	FUNCTION ATPGetRequest (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** atpSocket atpReqCount atpDataPtr	abOpcode (always tATPGetRequest) abResult atpAddress atpBitMap atpTransID atpActCount atpUserData atpXO
ATPSndRsp	Sends response to another socket	II	FUNCTION ATPSndRsp (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** atpSocket atpAddress atpRspBDSPtr atpTransID atpEOM atpNumBufs atpBDSSize	abOpcode (always tATPSdRsp) abResult
ATPAddRsp	Adds additional response packet to socket that has already been sent initial part of response	II	FUNCTION ATPAddRsp (abRecord:ABRecHandle) : OSErr;	abUserReference ** atpSocket atpAddress atpReqCount atpDataPtr atpTransID atpUserData atpEOM atpNumRsp	abOpcode (always tATPAddRsp) abResult
ATPResponse	Sends response to another socket but doesn't require BDS	II	FUNCTION ATPResponse (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** atpSocket atpAddress atpTransID atpRspUData atpRspBuf atpRspSize	abOpcode (always tATPResponse) abResult
ATPRspCancel	Dequeues ATPSndRsp or ATPResponse call	II	FUNCTION ATPRspCancel (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	Same as ATPSndRsp	Check abResult if error
NPBRegister	Adds name and address to node's name table	II	FUNCTION NBPRegister (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** nbpEntityPtr nbpBufPtr nbpBufSize nbpAddress.aSocket nbpRetransmitInfo	abOpcode (always tNPBRegister) abResult

(continued)

Table 3.302. Continued

Name	*Description*	*Defined **	*Pascal Definition*	*Prior to Call*	*On Return*
NPBLookup	Returns addresses of all entities with name	II	FUNCTION NBPLookup (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** nbpEntityPtr nbpBufPtr nbpBufSize nbpDataField nbpRetransmitInfo	abOpcode (always tNBPLookup) abResult nbpDataField
NBPExtract	Returns one address from NBPLookup list	II	FUNCTION NBPExtract (theBuffer:Ptr; numInBuf:INTEGER; whichOne:INTEGER; VAR abEntity:EntityName; VAR address:AddrBlock) : OSErr;	NOT USED	NOT USED
NBPConfirm	Confirms that entity still exists	II	FUNCTION NBPConfirm (abRecord:ABRecHandle; async:BOOLEAN) : OSErr;	abUserReference ** nbpEntityPtr nbpAddress nbpRetransmitInfo	abOpcode (always tNBPConfirm) abResult nbpDataField
OpenXPP	Opens XPP driver for use	V	FUNCTION OpenXPP (VAR xppRefnum: INTEGER) : OSErr;	NOT USED	not used
NBPRemove	Removes entity from table of names	II	FUNCTION NBPRemove (abEntity:EntityPtr) : OSErr;	NOT USED	not used
ASPOpenSession	Initiates workstation/server session	V	FUNCTION ASPOpenSession (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;		csCode sessRefnum aspTimeout aspRetry serverAddr scbPointer attnRoutine
ASPCloseSession	Closes workstation/server session	V	FUNCTION ASPCloseSession (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;		csCode sessRefnum
ASPAbortOS	Aborts pending ASPOpenSession	V	FUNCTION ASPAbortOS (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;		csCode abortSCBPointer
ASPGetParms	Returns three ASP parameters	V	FUNCTION ASPGetParms (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;		csCode aspMaxCmdSize aspQuantumSize numSesss
ASPCloseAll	Closes all active sessions	V	FUNCTION ASPCloseAll (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;		csCode
ASPUserWrite	Transfers data about a session	V	FUNCTION ASPUserWrite (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;	 rbSize wdSize	cmdResult csCode sessRefnum aspTimeout cbSize cbPtr rbSize rbPtr wdSize wdPtr ccbStart
ASPUserCommand	Sends command to server on a session	V	FUNCTION ASPUserCommand (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;	 rbSize	cmdResult csCode sessRefnum aspTimeout cbSize cbPtr rbSize rbPtr ccbStart

(continued)

Table 3.302. Continued

Name	Description	Defined *	Pascal Definition	Prior to Call	On Return
ASPGetStatus	Returns server status	V	FUNCTION ASPGetStatus (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;	rbSize	csCode aspTimeout aspRetry serverAddr rbSize rbPtr ccbStart
AFPCommand	Executes AFP command in XPPParmBlkPtr	V	FUNCTION AFPCommand (xParamBlock:XPPParmBlkPtr; async:BOOLEAN) : OSErr;	See source	See source
NBPLoad ***	Reads .NBP code from file to appplication heap	II	FUNCTION NBPLoad : OSErr;	NOT USED	NOT USED
NBPUnload ****	Marks NBP code as purgeable	II	FUNCTION NBPUnload : OSErr;	NOT USED	NOT USED
GetNodeAddress	Returns current node ID and network ID of caller	II	FUNCTION GetNodeAddress (VAR myNode:myNet:INTEGER) : OSErr;	NOT USED	NOT USED
IsMPPOpen	Returns TRUE if MPP driver is loaded and running	II	FUNCTION IsMPPOpen : BOOLEAN;	NOT USED	NOT USED
IsATPOpen	Returns TRUE if ATP driver is loaded and running	II	FUNCTION IsATPOpen : BOOLEAN;	NOT USED	NOT USED

Notes:
- These calls are the original high-level calls defined in Inside Macintosh Volume II. Volume V introduces new, so-called "preferred" calls for the low-level ALAP routines (see table below), which should be used in place of the original low-level routines.
- * Volume of Inside Macintosh where first definition of routine appeared (I–V).
- ** User-defined.
- *** On Macintosh 128K only; .ATP driver in application heap in Macintosh 128K, system heap for others.
- **** Has effect only on Macintosh 128K.

Source: Inside Macintosh, pages II-275 to II-304, V-516 to V-549

See Also: 3.156. Operating System Calls
3.303. to 3.317. other AppleTalk tables

New Low-Level (Preferred) ALAP Calls:

Old Routine	Description	New Name	Pascal Definition	Prior to Call	On Return
*	Builds BDS	BuildBDS	FUNCTION BuildBDS (buffPtr,bdsPtr:Ptr; buffSize:INTEGER) : INTEGER;		
*	Builds single-frame DDP write data structure	BuildDDPwds	PROCEDURE BuildDDPwds (wdsPtr,headerPtr,dataPtr:Ptr; destAddress:AddrBlock;DDPType:INTEGER; dataLen:INTEGER);		
*	Builds single-frame LAP write data structure	BuildLAPwds	PROCEDURE BuildLAPwds (wdsPtr,dataPtr:Ptr; destHost,protoType,frameLen:INTEGER);		
*	Returns current address of bridge in LO byte	GetBridgeAddress	FUNCTION GetBridgeAddress : INTEGER;		
NBPExtract	Extracts NBP entity names from look-up buffer	NBPExtract	FUNCTION NBPExtract (theBuffer:Ptr; numInBuf,whichOne: INTEGER; VAR abEntity:EntityName; VAR address:AddrBlock) : OSErr;		
*	Builds NBP entity structure	NBPSetEntity	PROCEDURE NBPSetEntity (buffer:Ptr; nbpObject,nbpType,nbpZone:Str32);		
*	Builds NBP names table entry	NBPSetNTE	PROCEDURE NBPSetNTE (ntePtr:Ptr; nbpObject,nbpType,nbpZone:Str32; Socket:INTEGER);		
AddResponse	Sends additional response packet to socket	PAddResponse	FUNCTION PAddResponse (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	userData csCode atpSocket atpFlags addrBlock reqLength reqPointer rspNum transID	
AttachPH	Adds protocol handler to node's protocol table	PAttachPH	FUNCTION PAttachPH (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode protType handler	
CloseATPSkt	Removes request receiving socket	PCloseATPSkt	FUNCTION PCloseATPSkt (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode atpSocket	

(continued)

Table 3.302. Continued

Old Routine	Description	New Name	Pascal Definition	Prior to Call	On Return
CloseSkt	Removes socket from table	PCloseSkt	FUNCTION PCloseSkt (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode socket	
ConfirmName	Confrims that entity still exists	PConfirmName	FUNCTION PConfirmName (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode interval count entityPtr confirmAddr	count newSocket
DetachPH	Removes handler from node's protocol table	PDetachPH	FUNCTION PDetachPH (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode protType	
GetRequest	Gets ready to receive request from SendRequest	PGetRequest	FUNCTION PGetRequest (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode atpSocket reqLength reqPointer	userData atpFlags addrBlock reqLength bitMap transID
*	Aborts specific GetRequest call	PKillGetReq	FUNCTION PKillGetReq (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode aKillQEl	
*	Aborts oustanding Lookup,Register,Confirm name	PKillNBP	FUNCTION PKillNBP (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode aKillQE1	
*	Aborts SendRequests/PNSendRequests	PKillSendReq	FUNCTION PKillSendReq (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode aKillQE1	
LoadNBP	Reads NBP code from system file on Mac 128K	NOT DEFINED	Assembly-language definition only; see source	csCode	
LookupName	Returns addresses of matches to name	PLookupName	FUNCTION PLookupName (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode interval count entityPtr retBuffPtr maxToGet	count numGotten
†	Specifies socket through which request is sent	PNSendRequest	FUNCTION PNSendRequest (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	userData csCode atpSocket atpFlags addrBlock reqLength reqPointer bdsPointer numOfBuffs timeOutVal retryCount	reqTID atpSocket atpFlags numOfResps retryCount intBuff
OpenATPSkt	Opens request receiving socket	POpenATPSkt	FUNCTION POpenATPSkt (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode atpSocket addrBlock	atpSocket
OpenSkt	Adds socket and listener to socket table	POpenSkt	FUNCTION POpenSkt (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode socket	listener socket
RegisterName	Adds name/address of entity to node's names table	PRegisterName	FUNCTION PRegisterName (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode interval count ntQElPtr verifyFlag	count
RelRspCB	Cancels SendResponse in exactly-once mode	PRelRspCB	FUNCTION PRelRspCB (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode atpSocket addrBlock	transID
RelTCB	Dequeues SendRequest call	PRelTCB	FUNCTION PRelTCB (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode addrBlock	transID
RemoveName	Removes name from node's table	PRemoveName	FUNCTION PRemoveName (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode entityPtr	

(continued)

Table 3.302. Continued

Old Routine	*Description*	*New Name*	*Pascal Definition*	*Prior to Call*	*On Return*
SendRequest	Sends request to and waits for response from socket	PSendRequest	FUNCTION PSendRequest (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	userData csCode atpFlags addrBlock reqLength reqPointer bdsPointer numOfBuffs timeOutVal retryCount	reqTID currBitMap atpFlags numOfResps retryCount
SendResponse	Sends response to socket	PSendResponse	FUNCTION PSendResponse (thePBptr:ATPPBPtr; async:BOOLEAN) : OSErr;	csCode atpSocket atpFlags addrBlock bdsPointer numOfBuffs bdsSize transID	reqTID userData
*	Enables/disables intranode delivery	PSetSelfSend	FUNCTION PSetSelfSend (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode newSelfFlag	oldSelfFlag
UnloadNBP	Marks NBP code as purgeable on Mac 128K	NOT DEFINED	Defined for assembly-language only; see source	csCode	
WriteDDP	Sends datagram to another socket	PWriteDDP	FUNCTION PWriteDDP (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode socket checksumFlag wdsPointer	
WriteLAP	Sends frame to another node	PWriteLAP	FUNCTION PWriteLAP (thePBptr:MPPPBPtr; async:BOOLEAN) : OSErr;	csCode wdsPointer	

Notes: * Not defined in Volume II; new low-level routine.

Source: Inside Macintosh, pages II-304 to II-345, V-511 to V-515

3.303. ABusRecord Data Structure

Size	*Name*	*Comments*
ABCallType	abOpcode	Call type
INTEGER	abResult	Result code from call
LONGINT	abUserReference	User-defined
Varies		

--> ***For lapProto:***

Size	*Name*	*Comments*
LAPAdrBlock	lapAddress	Destination or source node ID
INTEGER	lapReqCount	Length of frame or buffer, in bytes
INTEGER	lapActCount	Bytes actually received
Ptr	lapDataPtr	Pointer to frame or buffer

For ddpProto:

Byte	ddpType	Type of DDP protocol
Byte	ddpSocket	Socket number
AddrBlock	ddpAddress	Destination or source socket address
INTEGER	ddpReqCount	Length of datagram or buffer, in bytes
INTEGER	ddpActCount	Bytes actually received
Ptr	ddpDataPtr	Pointer to datagram or buffer
Byte	ddpNodeID	Original destination node ID

For nbpProto:

EntityPtr	nbpEntityPtr	Pointer to entity name
Ptr	nbpBufPtr	Pointer to buffer
INTEGER	nbpBufSize	Size of buffer, in bytes
INTEGER	nbpDataField	Socket number, of number of addresses
AddrBlock	nbpAddress	Socket address
RetransType	nbpRetransmitInfo	Retransmission information

For atpProto:

Byte	atpSocket	Socket number
AddrBlock	atpAddress	Destination or source socket address
INTEGER	atpReqCount	Size of buffer requested, in bytes
Ptr	atpDataPtr	Pointer to buffer
BDSPtr	atpRspBDSPtr	Pointer to BDS response
BitMapType	atpBitMap	Transaction bitmap
INTEGER	atpTransID	Transaction ID
INTEGER	atpActCount	Bytes actually received
LONGINT	atpUserData	User bytes
BOOLEAN	atpXO	Exactly-once flag
BOOLEAN	atpEOM	End-of-message flag
Byte	atpTimeOut	Retry timeout interval, in seconds
Byte	atpRetries	Maximum number of times to retry
Byte	atpNumBufs	Size of response or packets, in elements
Byte	atpNumRsp	Packets received, or sequence number
Byte	atpBDSSize	Size of BDS response, in elements
LONGINT	atpRspUData	User bytes in transaction message
Ptr	atpRspBuf	Pointer to response message buffer
INTEGER	atpRspSize	Response message buffer size, in bytes

Source: Inside Macintosh, pages II-231 to II-232, II-274 to II-299

See Also: 3.302. AppleTalk Routines
3.307. AddrBlock Data Structure
3.315. EntityName Data Structure

3.304. MPPParamBlock Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Type of queue
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine address
OSErr	ioResult	OS result code
StringPtr	ioNamePtr	ATP user bytes or command result
INTEGER	ioVRefNum	Volume reference or drive number
INTEGER	ioRefNum	Driver reference number
INTEGER	csCode	Call command code *
Varies		Based on MPPParmType

--> ***For LAPWriteParm:***

Size	*Name*	*Comments*
INTEGER	filler0	NOT USED
Ptr	wdsPointer	Pointer to write data structure

For AttachPHParm or DetachPHParm:

Byte	protType	ALAP protocol type
Byte	filler1	NOT USED
Ptr	handler	Pointer to protocol handler

For OpenSktParm, CloseSktParm, or WriteDDPParm:

Byte	socket	Socket number
Byte	checksumFlag	Checksum flag
Ptr	listener	Pointer to socket listener

For RegisterNameParm:

Byte	interval	Retry interval
Byte	count	Retry count
Ptr	entityPtr	Pointer to NTB element entity
Byte	verifyFlag	(set = verify)
Byte	filler3	NOT USED

For LookupNameParm:

Byte	interval	Retry interval
Byte	count	Retry count
Ptr	entityPtr	Pointer to NTB element entity
Ptr	retBuffPtr	Pointer to return buffer
INTEGER	retBuffSize	Size of return buffer, in bytes
INTEGER	maxToGet	Number of matches to get
INTEGER	numGotten	Number matched

For ConfirmNameParm:

Byte	interval	Retry interval
Byte	count	Retry count
Ptr	entityPtr	Pointer to NTB element entity
AddrBlock	confirmAddr	Pointer to entity
Byte	newSocket	Socket number
Byte	filler4	NOT USED

For SetSelfSendParm:

Byte	newSelfFlag	Self-send toggle flag
Byte	oldSelfFlag	Previous self-send state

For KillNBPParm:

Ptr	nKillQE1	Pointer to Qelem to cancel

Notes: * Automatically set.

Source: Inside Macintosh, pages V-511 to V-512

See Also: 3.302. AppleTalk Routines
3.307. AddrBlock Data Structure
3.315. EntityName Data Structure

3.305. XPPParamBlock Data Structure

Size	Name	Comments
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Type of queue
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine address
OSErr	ioResult	OS result code
LONGINT	cmdResult	Command result (ATP user bytes)
INTEGER	ioVRefNum	Volume reference or drive number
INTEGER	ioRefNum	Driver reference number
INTEGER	csCode	Call command code
Varies		Based on XPPParmType

--> ***For ASPAbortPrm:***

Size	Name	Comments
Ptr	abortSCBPtr	SCB pointer for Abort OS

For ASPSizeBlk:

INTEGER	aspMaxCmdSize	For SPGetParms
INTEGER	aspQuantumSize	For SPGetParms
INTEGER	numSesss	For SPGetParms

For XPPPrmBlk:

INTEGER	sessRefnum	Offset to session refnum
Byte	aspTimeout	Timeout value for ATP
Byte	aspRetry	Retry count for ATP
Varies		Based on XPPSubPrmType

<-- ***For ASPOpenPrm:***

Size	Name	Comments
AddrBlock	serverAddr	Address block for server
Ptr	scbPointer	Pointer to SCB
Ptr	attnRoutine	Pointer to attention routine

For ASPSubPrm:

INTEGER	cbSize	Command block size, in bytes
Ptr	cbPtr	Pointer to command block
INTEGER	rbSize	Reply buffer size, in bytes
Ptr	rbPtr	Pointer to reply buffer
Varies		Based upon XPPEndPrmType

--> ***For AFPLoginIPrm:***

Size	Name	Comments
AddrBlock	afpAddrBlock	AFPLogin address block
Ptr	afpSCBPtr	Pointer to SCB in AFPLogin
Ptr	afpAttnRoutine	Pointer to attn routine in AFPLogin

For ASPEndPrm:

INTEGER	wdSize	Write data size
Ptr	wdPtr	Pointer to write data
296 bytes	ccbStart	CCB memory

Source: Inside Macintosh, page V-535

See Also: 3.302. AppleTalk Routines
3.307. AddrBlock Data Structure

3.306. ALAP Frame

Size	Name	Comments
Byte	lapDstAdr	Destination node ID
Byte	lapSrcAdr	Source node ID
Byte	lapType	ALAP protocol type
0–600 bytes	lapData	Data
word	lapFrmChk	Frame check sequence

Notes: First three bytes constitute header.

Source: Inside Macintosh, page II-306

See Also: 3.302. AppleTalk Routines

3.307. AddrBlock Data Structure

Size	*Name*	*Comments*
INTEGER	aNet	Network number
Byte	aNode	Node ID
Byte	aSocket	Socket number

Source: Inside Macintosh, page II-281

See Also: 3.302. AppleTalk Routines

3.308. DDP Datagram Data Structure

Size	*Name*	*Comments*
3 bytes	ALAPHdr	See 3.306.
Word	ddpLength	*
Word	ddpChecksum	Checksum
Word	ddpDstNet	Destination network number
Word	ddpSrcNet	Source network number
Byte	ddpDstNode	Destination node ID
Byte	ddpSrcNode	Source node ID
Byte	ddpDstSkt	Destination socket number
Byte	ddpSrcSkt	Source socket number
Byte	ddpType	DDP protocol type
0–568 bytes	ddpData	

Notes: * LO 10 bits are datagram length, in bytes; HO 2 bits are not used; bits 10–13 represent hop count (number of bridges passed).

Source: Inside Macintosh, pages II-308 to II-309

See Also: 3.302. AppleTalk Routines
3.306. ALAP Frame

3.309. ATPParamBlock Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Type of queue
INTEGER	ioTrap	Routine trap
Ptr	ioCmdAddr	Routine address
ProcPtr	ioCompletion	Completion routine address
OSErr	ioResult	OS result code
LONGINT	userData	ATP user bytes
INTEGER	reqTID	Request transaction ID
INTEGER	ioRefNum	Driver reference number
INTEGER	csCode	Call command code *
Byte	atpSocket	Socket number (or currBitMap)
Byte	atpFlags	Control information
AddrBlock	addrBlock	Source/dest socket address
INTEGER	reqLength	Request/response length
Ptr	reqPointer	Pointer to request/response data
Ptr	bdsPointer	Pointer to response BDS
Varies		Based on MPPParmType

--> ***For SendRequestParm, NSendRequestParm:***

Size	*Name*	*Comments*
Byte	numOfBuffs	Number of buffers
Byte	timeOutVal	Timeout interval
Byte	numOfResps	Responses actually received
Byte	retryCount	Retries made
INTEGER	intBuff	USED INTERNALLY

For SendReponseParm:

Byte	filler0	Count of responses sent
Byte	BDSSize	Count of BDS elements
INTEGER	transID	Transaction ID

For GetRequestParm:

Byte	bitMap	Bitmap
Byte	filler1	NOT USED

For AddResponseParm:

Byte	rspNum	Sequence number
Byte	filler2	NOT USED

For KillSendReqParm, KillGetReqParm:

Ptr	aKillQE1	Ptr to queue element to kill

Notes: * Automatically set.

Source: Inside Macintosh, pages V-512 to V-513

See Also: 3.302. AppleTalk Routines
3.307. AddrBlock Data Structure
3.315. EntityName Data Structure

3.310. ATP Packet Data Structure

Size	*Name*	*Values/Comments*
3 bytes	ALAPHdr	See 3.306.
5 or 13 bytes	ddpDtgmHdr	See 3.308.
byte	atpControl	*
byte	atpBitMap or atpRespNo	Transaction bitmap or sequence #
word	atpTransID	Transaction ID
long	atpUserData	User bytes
0–578 bytes	atpData	Data

Notes: * Bits 6–7: Function code; Bit 5: XO; Bit 4: EOM; Bit 3: STS; Bits 0–2: NOT USED.

Source: Inside Macintosh, page II-313

See Also: 3.302. AppleTalk Routines
3.306. ALAP Frame
3.308. DDP Datagram Data Structure

3.311. ATP Flags Field

Bit Numbers

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Description*
X	X							NOT USED
		X						Exactly-once flag
			X					End-of-message flag
				X				Send-transmission-status flag
					X			NOT USED
						X		Transaction ID valid flag
							X	Send checksum flag

Source: Inside Macintosh, pages II-313 to II-314

See Also: 3.309. ATPParamBlock Data Structure

3.312. ATP Client Internet Address Data Structure

Size	*Name*	*Comments*
Word	netID	Network number
Byte	nodeID	Node ID
Byte	sktNumber	Socket number

Source: Inside Macintosh, page II-314

See Also: 3.309. ATPParamBlock Data Structure

3.313. NBP Packet Data Structure

Size	*Name*	*Comments*
3 bytes	alapFrameHdr	ALAP frame header (see 3.306.)
5 or 13 bytes	ddpDatagramHdr	DDP datagram header (see 3.308.)
Byte	packType/tupleCount *	Packet type and tuple count
Byte	nbpPackID	NBP packet ID
Varies	tuples	Tuple list

Notes: * Packet type is upper 4 bits, tuple count is lower 4 bits.

Source: Inside Macintosh, page II-320

See Also: 3.302. AppleTalk Routines
3.306. ALAP Frame
3.308. DDP Datagram Data Structure

3.314. Names Table Entry Data Structure

Size	*Name*	*Comments*
Ptr	ntLink	Pointer to next entry
Word	ntNetNumber	Network number
Byte	ntNodeID	Node ID
Byte	ntSocket	Socket number
Byte	ntReserved	USED INTERNALLY
Byte	ntEntityLength	Length of entity name
Varies	ntEntityName	Entity name
Byte	ntEntityTypeLength	Length of entity type
Varies	ntEntityTypeName	Entity type
Byte	ntEntityZoneLength	Length of entity zone
Varies	ntEntityZoneName	Entity zone

Source: Inside Macintosh, page II-321

See Also: 3.302. AppleTalk Routines

3.315. EntityName Data Structure

Size	*Name*	*Comments*
Str32	objStr	Entity object (name)
Str32	typeStr	Entity type
Str32	zoneStr	Entity zone

Source: Inside Macintosh, page II-298

See Also: 3.302. AppleTalk Routines
3.314. Names Table Entry Data Structure

3.316. RetransType Data Structure

Size	*Name*	*Comments*
Byte	retransInterval	Interval to retransmit, in 8-tick units
Byte	retransCount	Number of retrans attempts

Source: Inside Macintosh, page II-298

See Also: 3.302. AppleTalk Routines

3.317. AppleTalk Result Codes

Value	*Constant*	*Comments*
-3109	sktClosedErr	General: async call aborted because socket closed
-3108	recNotFnd	General: ABRecord not found
-3107	atpBadRsp	ATP: response from ATPRequest was bad
-3106	atpLenErr	ATP: response message too large
-3105	readQErr	General: socket or protocol type invalid or not found
-3104	extractErr	NBP: can't find tuple in buffer
-3103	cksumErr	DDP: bad checksum
-3102	noMPPError	MPP: driver not installed
-3101	buf2SmallErr	ALAP: frame too large for buffer, or DDP: datagram too large for buffer
-1105	reqAborted	General: request aborted
-1104	noDataArea	ATP: too many outstanding calls
-1103	noSendResp	ATP: ATPAddRsp before ATPSndRsp
-1102	cbNotFound	ATP: control block not found
-1101	noRelErr	ATP: no release received
-1100	badBuffNum	ATP: bad sequence number
-1099	badATPSkt	ATP: bad responding socket
-1098	tooManySkts	ATP: too many responding sockets
-1097	tooManyReqs	ATP: too many concurrent requests
-1096	reqFailed	ATP: ATPSndReqest failed because retry count was exceeded
-1029	nbpNISErr	NBP: names information socket error
-1028	nbpNotFound	NBP: name not found
-1027	nbpDuplicate	NBP: duplicate name, already exists
-1026	nbpConfDiff	NBP: name confirmed for different socket
-1025	nbpNoConfirm	NBP: name not confirmed
-1024	nbpBuffOvr	NBP: buffer overflow
-98	portNotCf	General: driver open error, port not configured correctly
-97	portInUse	General: driver open error, port already in use
-95	excessCollsns	ALAP: no CTS after 32 RTS's
-94	lapProtErr	ALAP: error attaching or detaching ALAP protocol type
-93	noBridgeErr	General: no bridge found
-92	ddpLenErr	DDP: datagram length too big, or ALAP: data length too big
-91	ddpSktErr	DDP: socket error
0	noErr	No error

Source: Inside Macintosh, pages II-337 to II-338

See Also: 3.302. AppleTalk Routines

3.318. Disk Initialization Routines

Name	*Description*	*Defined **	*Pascal Definition*
DIBadMount	Called when disk-inserted event indicates error	II	FUNCTION DIBadMount (where:Point; evtMessage:LONGINT) : INTEGER;
DIFormat	Formats disk in drive	II	FUNCTION DIFormat (drvNum:INTEGER) : OSErr;
DILoad	Loads Disk Init package into unpurgeable memory	II	PROCEDURE DILoad;
DIUnload	Makes Disk Init package purgeable	II	PROCEDURE DIUnload;
DIVerify	Verifies format of disk	II	FUNCTION DIVerify (drvNum:INTEGER) : OSErr;
DIZero	Writes volume info and other init information	II	FUNCTION DIZero (drvNum:INTEGER; volName:Str255) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages II-396 to II-399

See Also: 3.156. Operating System Calls
3.319. Disk Initialization Routine Selectors
3.320. Disk Initialization Result Codes

3.319. Disk Initialization Routine Selectors

Value	*Constant*
0	diBadMount
2	diLoad
4	diUnload
6	diFormat
8	diVerify
10	diZero

Notes: Used in assembly language to call packet routines.

Source: Inside Macintosh, page II-400

See Also: 3.318. Disk Initialization Routines

3.320. Disk Initialization Result Codes

Value	*Constant*	*Comments*
-108	memFullErr	Not enough room in heap zone
-84	firstDskErr	First of range of low-level disk errors
-64	lastDskErr	Last of range of low-level disk errors
-60	badMDBErr	Master directory block bad
-58	extFSErr	External file system error
-57	noMacDskErr	Not a Macintosh disk
-56	nsDrvErr	Drive not found
-55	volOnLinErr	Volume already on line
-50	paramErr	Drive number bad
-36	ioErr	I/O error
0	noErr	No error

Source: Inside Macintosh, page II-400

See Also: 3.318. Disk Initialization Routines

3.321. Time Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
InsTime	Adds task to Time Manager queue	IV	PROCEDURE InsTime (tmTaskPtr:QElemPtr);
PrimeTime	Schedules routine to be executed	IV	PROCEDURE PrimeTime (tmTaskPtr,count:LONGINT);
RmvTime	Removes task from Time Manager queue	IV	PROCEDURE RmvTime (tmTaskPtr:QElemPtr);

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, pages IV-300 to IV-301

See Also: 3.156. Operating System Calls
3.322. TMTask Data Structure

3.322. TMTask Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
ProcPtr	tmAddr	Pointer to time routine
INTEGER	tmCount	RESERVED

Source: Inside Macintosh, page IV-299

See Also: 3.321. Time Manager Routines

3.323. Deferred Task Manager Routines

Name	*Description*	*Defined **	*Pascal Definition*
DTInstall	Adds task to deferred task queue	V	FUNCTION DTInstall (dtTaskPtr:QElemPtr) : OSErr;

Notes: * Volume of Inside Macintosh where first definition of routine appeared (I–V).

Source: Inside Macintosh, page V-467

See Also: 3.156. Operating System Calls
3.324. DeferredTask Data Structure

3.324. DeferredTask Data Structure

Size	*Name*	*Comments*
QElemPtr	qLink	Pointer to next queue entry
INTEGER	qType	Queue type
INTEGER	dtFlags	RESERVED
ProcPtr	dtAddr	Pointer to task
LONGINT	dtParm	Optional parameter for task
LONGINT	dtReserved	RESERVED (must be 0)

Source: Inside Macintosh, page V-466

See Also: 3.323. Deferred Task Manager Routines

3.325. SANE Routines

Name	*Description*	*Pascal Definition*
Annuity	Returns annuity (1-(1+r)^(-n))r	FUNCTION Annuity (r,n:extended) : extended;
ArcTan	Returns arctangent	FUNCTION ArcTan (x:extended) : extended;
ClassComp	Returns class of comp	FUNCTION ClassComp (x:comp) : NumClass;
ClassDouble	Returns class of double	FUNCTION ClassDouble (x:double) : NumClass;
ClassExtended	Returns class of extended	FUNCTION ClassExtended (x:extended) : NumClass;
ClassReal	Returns class of single	FUNCTION ClassReal (x:real) : NumClass;
Compound	Returns (1 + r)^n	FUNCTION Compound (r,n:extended) : extended;
CopySign	Copies sign of first argument onto second	FUNCTION CopySign (x,y:extended) : extended;
Cos	Returns cosine	FUNCTION Cos (x:extended) : extended;
Dec2Num	Converts decimal record to SANE number	FUNCTION Dec2Num (d:Decimal) : extended;
Dec2Str	Converts decimal number to decimal string	PROCEDURE Dec2Str (f:DecForm; d:decimal; VAR s:DecStr);
Exp	Returns base-e exponential e^x	FUNCTION Exp (x:extended) : extended;
Exp1	Returns exp(x)-1	FUNCTION Exp1 (x:extended) : extended;
Exp2	Returns base-2 exponential	FUNCTION Exp2 (x:extended) : extended;
GetEnvironment	Returns environment word	PROCEDURE GetEnvironment (VAR e:Environment);
GetHaltVector	Returns halt vector	FUNCTION GetHaltVector : LONGINT;
GetPrecision	Returns arithmetic precision	FUNCTION GetPrecision : RoundPre;
GetRound	Returns rounding direction	FUNCTION GetRound : RoundDir;
Ln	Returns base-e logarithm	FUNCTION Ln (x:extended) : extended;
Ln1	Returns n(1+x)	FUNCTION Ln1 (x:extended) : extended;
Log2	Returns base-2 logarithm	FUNCTION LOG2 (x:extended) : extended;
Logb	Returns binary exponent of x	FUNCTION Logb (x:extended) : extended;
NextDouble	Returns value x value towards y	FUNCTION NextDouble (x,y:double) : double;
NextExtended	Returns value x value towards y	FUNCTION NextExtended (x,y:extended) : extended;
NextReal	Returns value x value towards y	FUNCTION NextReal (x,y:real) : real;
Num2Comp	Converts number to SANE comp format	FUNCTION Num2Comp (x:extended) : comp;
Num2Dec	Converts SANE number to decimal number	PROCEDURE Num2Dec (f:DecForm; x:extended; VAR d:Decimal);
Num2Double	Converts number to SANE double format	FUNCTION Num2Double (x:extended) : double;
Num2Integer	Converts number to integer format	FUNCTION Num2Integer (x:extended) : INTEGER;
Num2Longint	Converts number to integer format	FUNCTION Num2Longint (x:extended) : LONGINT;
Num2Real	Converts number to SANE single format	FUNCTION Num2Real (x:extended) : real;
Num2Str	Converts number to string	PROCEDURE Num2Str (f:DecForm; x:extended; VAR s:DecStr);
ProcEntry	Returns current environment, sets default	PROCEDURE ProcEntry (VAR e:Environment);
ProcExit	Restores caller's environment	PROCEDURE ProcExit (e:Environment);
RandomX	Returns pseudo random number	FUNCTION RandomX (VAR x:extended) : extended;
Relation	Returns relation of two values	FUNCTION Relation (x,y:extended) : RelOp;
Remainder	Returns remainder of division	FUNCTION Remainder (x,y:extended; VAR quo:INTEGER) : extended;
Rint	Returns rounded integer	FUNCTION Rint (x:extended) : extended;
Scalb	Returns x * 2^n	FUNCTION Scalb (n:integer; x:extended) : extended;
SetEnvironment	Sets environment word	PROCEDURE SetEnvironment (e:Environment);
SetException	Sets exception flags	PROCEDURE SetException (e:Exception; b:BOOLEAN);
SetHalt	Sets halt settings	PROCEDURE SetHalt (e:Exception; b:BOOLEAN);
SetHaltVector	Sets halt vector	PROCEDURE SetHaltVector (v:LONGINT);
SetPrecision	Sets arithmetic precision	PROCEDURE SetPrecision (p:RoundPre);
SetRound	Sets rounding direction	PROCEDURE SetRound (r:RoundDir);
Sin	Returns sine	FUNCTION Sin (x:extended) : extended;
Str2Dec	Converts decimal string to decimal number	PROCEDURE Str2Dec (s:DecStr; VAR Index:integer; VAR d:Decimal; VAR ValidPrefix: BOOLEAN);
Str2Num	Converts string to extended number	FUNCTION Str2Num (s:DecStr) : extended;
Tan	Returns tangent	FUNCTION Tan (x:extended) : extended;
TestException	Returns exception flags	FUNCTION TestException (e:Exception) : BOOLEAN;
TestHalt	Returns halt settings	FUNCTION TestHalt (e:Exception) : b:BOOLEAN;
Xpwrl	Returns integer exponential x^i	FUNCTION Xpwrl (x:extended; i:INTEGER) : extended;
XpwrY	Returns exponential x^y	FUNCTION XpwrY (x, y:extended) : extended;

Notes: These definitions are suggested only; actual interface is dependent upon compiler.

Source: Apple Numerics Manual (second edition), pages 22 to 68

See Also:
3.326. SANE Data Types
3.327. SANE Not-a-Number Codes
3.328. Comparison Effect on 68000 Status Bits
3.329. Result Codes from SANE Inquiries
3.330. SANE Environment Word
3.331. Test-Exception Word
3.332. Exception Halt Stack Frame

3.326. SANE Data Types

Single

Component	s	e	f
Bit #s	31	30 – 23	22 – 0

Value translates as follows:

Value of e	Value of f	Translated Value	Class of Value
0 < e < 255	IGNORED	(-1)^s x 2^(e-127) x (1.f)	Normalized
e = 0	f <> 0	(-1)^s x 2^(-126) x (0.f)	Denormalized
e = 0	f = 0	(-1)^s x 0	Zero
e = 255	f = 0	(-1)^s x Infinity	Infinity
e = 255	f < > 0	NaN	NaN

Double

Component	s	e	f
Bit #s	63	62 – 51	50 – 0

Value translates as follows:

Value of e	Value of f	Translated Value	Class of Value
0 < e < 2047	IGNORED	(-1)^s x 2^(e-1023) x (1.f)	Normalized
e = 0	f <> 0	(-1)^s x 2^(-1022) x (0.f)	Denormalized
e = 0	f = 0	(-1)^s x 0	Zero
e = 2047	f = 0	(-1)^s x Infinity	Infinity
e = 2047	f < > 0	NaN	NaN

Comp

Component	s	d
Bit #s	63	62 – 0

Value translates as follows:

Value of s	Value of d	Translated Value
1	0	Unique comp NaN
1	<> 0	Two's complement of 64 bits
0	IGNORED	Two's complement of 64 bits

Extended

Component	s	e	i	f
Bit #s	79	78 – 63	62	0

or

Component	s	e	NOT USED	i	f
Bit #s	95	94 – 79	78 – 63	62	0

Value translates as follows:

Value of e	Value of i	Value of f	Translated Value	Class of Value
0 ≤ e ≤ 32767	1	IGNORED	(-1)^s x 2^(e-16383) x (1.f)	Normalized
0 ≤ e ≤ 32766	0	f <> 0	(-1)^s x 2^(e-16383) x (0.f)	Denormalized
0 ≤ e ≤ 32767	0	f = 0	(-1)^s x 0	Zero
e = 32767	IGNORED	f = 0	(-1)^s x Infinity	Infinity
e = 32767	IGNORED	f < > 0	NaN	NaN

Source: Apple Numerics Manual (second edition), pages 16 to 18

See Also: 3.325. SANE Routines

3.327. SANE Not-a-Number Codes

Value	*Name*	*Comments*
$1	NANSQRT	Invalid square root (example: sqrt(-1))
$2	NANADD	Invalid addition (example: (+INF) - (+INF))
$4	NANDIV	Invalid division (example: 0/0)
$8	NANMUL	Invalid multiplication (example: 0 x INF)
$9	NANREM	Invalid remainder, invalid mod (example: x rem 0)
$11	NANASCBIN	Invalid ASCII string (example: 45N34)
$14	NANCOMP	Converted comp NaN to floating
$15	NANZERO	Attempted to create NaN with zero code
$21	NANTRIG	Invalid argument to trigonometric routine
$22	NANINVTRIG	Invalid argument to inverse trigonometric routine
$24	NANLOG	Invalid argument to log routine
$25	NANPOWER	Invalid argument to x^i or x^y routine
$26	NANFINAN	Invalid argument to financial function
$FF	NAN68881	NAN created by 68881

Source: Apple Numerics Manual (second edition), page 41

See Also: 3.325. SANE Routines

3.328. Comparison Effect on 68000 Status Bits

Status Bit					
X	*N*	*Z*	*V*	*C*	*Result of Comparison*
0	0	0	0	0	Greater than
1	1	0	0	1	Less than
0	0	1	0	0	Equal to
0	0	0	1	0	Unordered

Source: Apple Numerics Manual (second edition), page 152

See Also: 3.325. SANE Routines

3.329. Result Codes from SANE Inquiries

Value	*Class*
1	Signaling NaN
2	Quiet NaN
3	Infinity
4	Zero
5	Normalized
6	Denormalized

Notes: The sign of the result is the sign of the source; magnitude gives class value, as above.

Source: Apple Numerics Manual (second edition), page 153

See Also: 3.325. SANE Routines

3.330. SANE Environment Word

Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	*Description*	*Values*
X																NOT USED	
	X	X														Rounding direction	00 = to nearest, 01 = to upward, 10 = downward, 11 = toward zero
			X													Inexact exception flag	
				X												Divide-by-zero exception flag	
					X											Overflow exception flag	
						X										Underflow exception flag	
							X									Invalid exception flag	
								X								NOT USED	
									X	X						Rounding precision	00 = extended, 01 = double, 10 = single, 11 = undefined
											X					Inexact halt enable	
												X				Divide-by-zero halt enable	
													X			Overflow halt enable	
														X		Underflow halt enable	
															X	Invalid halt enable	

Source: Apple Numerics Manual (second edition), pages 162 to 163

See Also: 3.325. SANE Routines

3.331. Test-Exception Word

Bit Numbers

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Description*
X	X	X	X	X	X	X	X	X	X	X						NOT USED
											X					Inexact exception
												X				Divide-by-zero exception
													X			Overflow exception
														X		Underflow exception
															X	Invalid exception

Source: Apple Numerics Manual, page 164

See Also: 3.325. SANE Routines

3.332. Exception Halt Stack Frame

Size	*Description*
Long	Pointer to MISC record
Long	SRC2 address
Long	SRC address
Long	DST address
Word	Opcode
Long	Return address

--->

MISC Record

Long	Pending D0
Word	Pending CCR
Word	Halt exceptions

Source: Apple Numerics Manual (second edition), page 169

See Also: 3.325. SANE Routines

SECTION IV

HARDWARE

4.01. Macintosh Machine Summary

	128K	*512K*	*Plus*	*512KE*	*SE*	*II*
CPU	7.8Mhz 68000	7.8Mhz 68000	7.8Mhz 68000	7.8Mhz 68000	7.8Mhz 68000	15.7Mhz 68020
Coprocessor(s)	None	None	None	None	None	68881 math
RAM Memory Size	128K soldered in	512K soldered in	1MB uses SIMMs *	512K soldered in	1MB uses SIMMs *	1MB uses SIMMs **
ROM Memory Size	64K	64K	128K ***	128K	256K	256K ****
Memory Addressing	24 bit	24 bit	24 bit	24 bit	24 bit	24 or 32 bit
Floppy Disk(s)	One 400K †	One 400K †	One 800K †	One 800K †	One 800K †	One 800K ¥
Hard Disk	None	None	External optional	None	Internal 20MB optional	Internal 40MB optional
Ports	Two serial	Two serial	Two serial, one SCSI	Two serial	Two serial, one SCSI, two ADB	Two serial, one SCSI, two ADB
Expansion Slots	None	None	None	None	One	Six (NuBus)
Sound	Four-voice chip	Four-voice chip	Four-voice chip	Four-voice chip	Four-voice chip	Apple Sound Chip
Keyboard	No arrow keys, keypad	No arrow keys, keypad	Includes arrow keys, keypad	No arrow keys, keypad	Optional §	Optional §
Display	9" mono 512x342	9" mono 512x342	9" mono 512x342	9" mono 512x342	9" mono 512x342	Optional 640x480 ‡
Height	13.5"	13.5"	13.5"	13.5"	13.5"	5.5"
Depth	10.9"	10.9"	10.9"	10.9"	10.9"	14.8"
Width	9.7"	9.7"	9.7"	9.7"	9.7"	18.7"
Weight	16 lbs 8 oz	16 lbs 8 oz	16 lbs 8 oz	16 lbs 8 oz	17 lbs +	24 lbs +
Software Supplied	MacPaint, MacWrite, Finder	MacPaint, MacWrite, Finder	Finder, HyperCard	Finder	Finder,HyperCard	Finder, HyperCard
Introduced	Mar-84	Sep-84	Jan-86	Jan-86	Jan-87	Apr-87

Notes:

- * Expandable to 4MB using 1MB SIMMs instead of supplied 256K SIMMs.
- ** Expandable to 8MB using 1MB SIMMs instead of supplied 256K SIMMs.
- *** Expandable to 256K.
- **** Expandable to 512K.
- † Plus optional external drive.
- ¥ Plus optional internal drive.
- § Actually not optional, but choice of two PC-like keyboards.
- ‡ Apple's base option: 13" mono or 13" color monitors.

Source: Technical Introduction to the Macintosh Family, pages 9 to 12, 251 to 255

See Also: 2.077. System Software History

4.02. Macintosh 128/512/Plus Keyboard Command Protocol

Command Value (from computer)	*Command*	*Response Value (from keyboard)*
$10	INQUIRY	Key code or Null ($7B)
$14	INSTANT	Key code or Null ($7B)
$16	MODEL NUMBER	Model code *
$36	TEST	ACK ($7D) or NAK ($77)

Notes: * Model code is formatted as follows:

Bit #	*Meaning*
7	1 = another device is attached
4–6	Next device number (1–8)
1–3	Keyboard model number (1–8)
0	Always 1

Source: Inside Macintosh, pages III-30 to III-31

See Also: 4.03. Macintosh 128/512/Plus Key Transition Codes

4.03. Macintosh 128/512/Plus Key Transition Codes

Key	*Keydown Code*	*Keyup Code*
`	$65	$E5
1	$25	$A5
2	$27	$A7
3	$29	$A9
4	$2B	$AB
5	$2F	$AF
6	$2D	$AD
7	$35	$B5
8	$39	$B9
9	$33	$B3
0	$3B	$BB
-	$37	$B7
=	$31	$B1
[Delete]	$67	$E7
[Tab]	$61	$E1
Q	$19	$99
W	$1B	$9B
E	$1D	$9D
R	$1F	$9F
T	$23	$A3
Y	$13	$93
U	$41	$C1
I	$45	$C5
O	$3F	$BF
P	$47	$C7
[	$43	$C3
]	$3D	$BD
\	$55	$D5
[Caps lock]	$73	$F3
A	$01	$81
S	$03	$83
D	$05	$85
F	$07	$87
G	$0B	$8B
H	$09	$89
J	$4D	$CD
K	$51	$D1
L	$4B	$CB
;	$53	$D3
'	$4F	$CF

Key	*Keydown Code*	*Keyup Code*
[Return]	$49	$C9
[Left shift]	$71	$F1
Z	$0D	$8D
X	$0F	$8F
C	$11	$91
V	$13	$93
B	$17	$97
N	$5B	$DB
M	$5D	$DD
,	$57	$D7
.	$5F	$DF
/	$59	$D9
[Right shift]	$71	$F1
[Left option]	$75	$F5
[Command]	$6F	$EF
[Space]	$63	$E3
[Enter]	$69	$E9
[Right option]	$75	$F5
[Up arrow]	$1B	$9B
[Left arrow]	$0D	$8D
[Right arrow]	$05	$85
[Down arrow]	$11	$91
[Clear]	$0F	$8F
Keypad = (*)	$11	$91
Keypad / (*)	$1B	$9B
Keypad * (*)	$05	$85
Keypad 7	$33	$B3
Keypad 8	$37	$B7
Keypad 9	$39	$B9
Keypad -	$1D	$9D
Keypad 4	$2D	$AD
Keypad 5	$2F	$AF
Keypad 6	$31	$B1
Keypad + (*)	$0D	$8D
Keypad 1	$27	$A7
Keypad 2	$29	$A9
Keypad 3	$2B	$AB
Keypad enter	$19	$99
Keypad 0	$25	$A5
Keypad .	$03	$83

Notes:
- Key definitions shown are for U.S. keyboard.
- These codes are not the ones returned by the keyboard software driver. (See tables listed below for those codes.)
- * Not on original keypad. (Original used arrow keys in these spots.)

Source: Inside Macintosh, pages III-31 to III-32, IV-250

See Also:
4.04. Macintosh U.S. Key Codes
4.05. Macintosh International Key Codes
4.06. Macintosh ADB Standard Key Codes
4.07. Macintosh ADB Extended Key Codes

4.04. Macintosh U.S. Key Codes

Key	128/512 Code	Plus Code
`	$32	$32
1	$12	$12
2	$13	$13
3	$14	$14
4	$15	$15
5	$17	$17
6	$16	$16
7	$1A	$1A
8	$1C	$1C
9	$19	$19
0	$1D	$1D
-	$1B	$1B
=	$18	$18
[Delete]	$33	$33
[Tab]	$30	$30
Q	$0C	$0C
W	$0D	$0D
E	$0E	$0E
R	$0F	$0F
T	$11	$11
Y	$10	$10
U	$20	$20
I	$22	$22
O	$1F	$1F
P	$23	$23
[	$21	$21
]	$1E	$1E
\	$2A	$2A
[Caps lock]	$39	$39
A	$00	$00
S	$01	$01
D	$02	$02
F	$03	$03
G	$05	$05
H	$04	$04
J	$26	$26
K	$28	$28
L	$25	$25
;	$29	$29
'	$27	$27

Key	128/512 Code	Plus Code
[Return]	$24	$24
[Left shift]	$38	$38
Z	$06	$06
X	$07	$07
C	$08	$08
V	$09	$09
B	$0B	$0B
N	$2D	$2D
M	$2E	$2E
,	$2B	$2B
.	$2F	$2F
/	$2C	$2C
[Right shift]	$38	$38
[Left option]	$3A	$3A
[Command]	$37	$37
[Space]	$31	$31
[Enter]	$34	NA
[Right option]	$3A	NA
[Left arrow]	NA	$46
[Right arrow]	NA	$42
[Up arrow]	NA	$4D
[Down arrow]	NA	$48
Keypad clear	NA	$47
Keypad =	NA	$48
Keypad /	NA	$4D
Keypad *	NA	$42
Keypad 7	NA	$59
Keypad 8	NA	$5B
Keypad 9	NA	$5C
Keypad +	NA	$46
Keypad 4	NA	$56
Keypad 5	NA	$57
Keypad 6	NA	$58
Keypad -	NA	$4E
Keypad 1	NA	$53
Keypad 2	NA	$54
Keypad 3	NA	$55
Keypad enter	NA	$4C
Keypad 0	NA	$52
Keypad .	NA	$41

Notes: These codes are not the key down and key up codes sent to the computer by the keyboard.

Source: Inside Macintosh, pages I-251, V-191

See Also: 4.03. Macintosh 128/512/Plus Key Transition Codes
4.05. Macintosh International Key Codes
4.06. Macintosh ADB Standard Key Codes
4.07. Macintosh ADB Extended Key Codes

4.05. Macintosh International Key Codes

Key	*Code*
§	$32
1	$12
2	$13
3	$14
4	$15
5	$17
6	$16
7	$1A
8	$1C
9	$19
0	$1D
-	$1B
=	$18
[Delete]	$33
[Tab]	$30
Q	$0C
W	$0D
E	$0E
R	$0F
T	$11
Y	$10
U	$20
I	$22
O	$1F
P	$23
[	$21
]	$1E
[Return]	$2A
[Caps lock]	$39

Key	*Code*
A	$00
S	$01
D	$02
F	$03
G	$05
H	$04
J	$26
K	$28
L	$25
;	$29
'	$27
`	$24
[Left shift]	$38
\	$06
Z	$07
X	$08
C	$09
V	$0B
B	$2D
N	$2E
M	$2B
,	$2F
.	$2C
/	$0A
[Right shift]	$38
[Left option]	$3A
[Command]	$37
[Space]	$31
[Enter]	$34
[Right option]	$3A

Notes: • These codes are not the key down and key up codes sent to the computer by the keyboard.
• Keys shown are for the British version of the keyboard.

Source: Inside Macintosh, page I-251

See Also: 4.03. Macintosh 128/512/Plus Key Transition Codes
4.04. Macintosh U.S. Key Codes
4.06. Macintosh ADB Standard Key Codes
4.07. Macintosh ADB Extended Key Codes

4.06. Macintosh ADB Standard Key Codes

Key	*Code*
[Escape]	$35
1	$12
2	$13
3	$14
4	$15
5	$17
6	$16
7	$1A
8	$1C
9	$19
0	$1D
-	$1B
=	$18
[Delete]	$33
[Tab]	$30
Q	$0C
W	$0D
E	$0E
R	$0F
T	$11
Y	$10
U	$20
I	$22
O	$1F
P	$23
[	$21
]	$1E
\	$2A
[Control]	$3B
A	$00
S	$01
D	$02
F	$03
G	$05
H	$04
J	$26
K	$28
L	$25
;	$29
'	$27

Key	*Code*
[Return]	$24
[Left shift]	$38
Z	$06
X	$07
C	$08
V	$09
B	$0B
N	$2D
M	$2E
,	$2B
.	$2F
/	$2C
[Right shift]	$38
[Caps lock]	$39
[Option]	$3A
[Command]	$37
[Space]	$31
[Enter]	NA
[Right option]	NA
[Left arrow]	$7B
[Right arrow]	$7C
[Up arrow]	$7E
[Down arrow]	$7D
Keypad clear	$47
Keypad =	$51
Keypad /	$4B
Keypad *	$43
Keypad 7	$59
Keypad 8	$5B
Keypad 9	$5C
Keypad +	$45
Keypad 4	$56
Keypad 5	$57
Keypad 6	$58
Keypad -	$4E
Keypad 1	$53
Keypad 2	$54
Keypad 3	$55
Keypad enter	$4C
Keypad 0	$52
Keypad .	$41

Notes: These codes are not the key down and key up codes sent to the computer by the keyboard.

Source: Inside Macintosh, page V-191

See Also: 4.03. Macintosh 128/512/Plus Key Transition Codes
4.04. Macintosh U.S. Key Codes
4.05. Macintosh International Key Codes
4.07. Macintosh ADB Extended Key Codes

4.07. Macintosh ADB Extended Key Codes

Key	*Code*
[Escape]	$35
[f1]	$7A
[f2]	$78
[f3]	$63
[f4]	$76
[f5]	$60
[f6]	$61
[f7]	$62
[f8]	$64
[f9]	$65
[f10]	$6D
[f11]	$67
[f12]	$6F
[f13]	$69
[f14]	$6B
[f15]	$71
`	$32
1	$12
2	$13
3	$14
4	$15
5	$17
6	$16
7	$1A
8	$1C
9	$19
0	$1D
-	$1B
=	$18
[Delete]	$33
[Tab]	$30
Q	$0C
W	$0D
E	$0E
R	$0F
T	$11
Y	$10
U	$20
I	$22
O	$1F
P	$23
[	$21
]	$1E
\	$2A
[Caps lock]	$39
A	$00
S	$01
D	$02
F	$03
G	$05
H	$04
J	$26

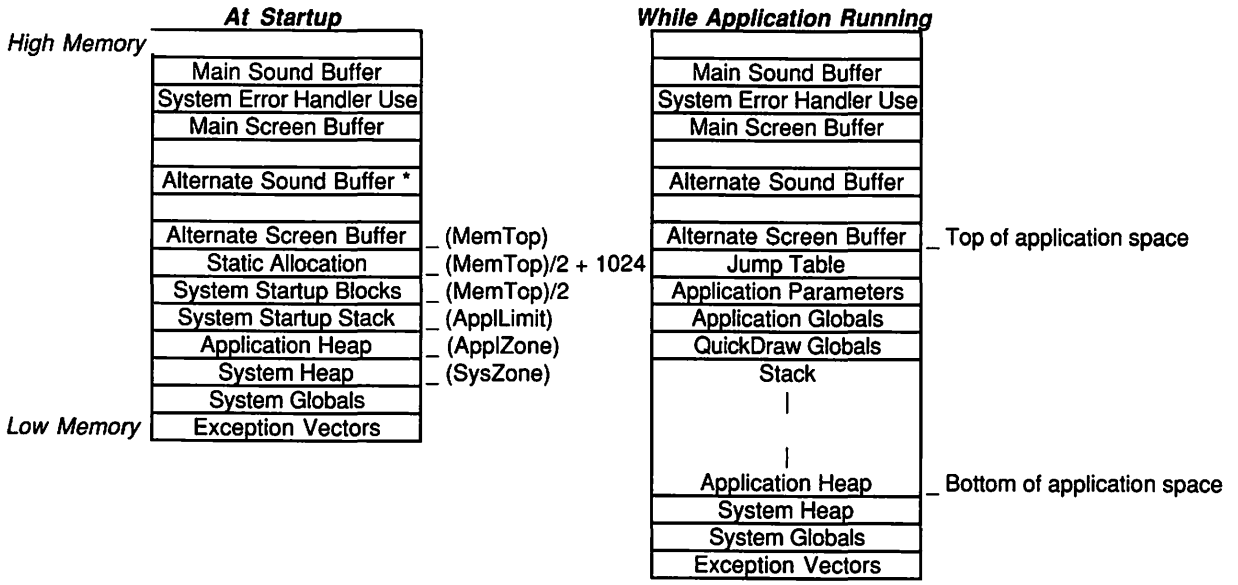
Key	*Code*
K	$28
L	$25
;	$29
"	$27
[Return]	$24
[Left shift]	$38
Z	$06
X	$07
C	$08
V	$09
B	$0B
N	$2D
M	$2E
,	$2B
.	$2F
/	$2C
[Right shift]	$38 ($3C)
[Control]	$3B
[Left option]	$3A
[Left command]	$37
[Space]	$31
[Right command]	$37
[Right option]	$3A ($3D)
[Right control]	$36 ($3E)
[Left arrow]	$7B
[Right arrow]	$7C
[Up arrow]	$7E
[Down arrow]	$7D
Keypad clear	$47
Keypad =	$51
Keypad /	$4B
Keypad *	$43
Keypad 7	$59
Keypad 8	$5B
Keypad 9	$5C
Keypad +	$45
Keypad 4	$56
Keypad 5	$57
Keypad 6	$58
Keypad -	$4E
Keypad 1	$53
Keypad 2	$54
Keypad 3	$55
Keypad enter	$4C
Keypad 0	$52
Keypad .	$41
[Help]	$72
[Home]	$73
[Page up]	$74
[Del]	$75
[End]	$77
[Page down]	$79

Notes: • These codes are not the key down and key up codes sent to the computer by the keyboard.
• Numbers in parentheses indicate values when key reassignment is made.

Source: Inside Macintosh, page V-192

See Also: 4.03. Macintosh 128/512/Plus Key Transition Codes
4.04. Macintosh U.S. Key Codes
4.05. Macintosh International Key Codes
4.06. Macintosh ADB Standard Key Codes

4.08. Macintosh 128/512/Plus/SE System RAM Layout



Notes: • Names in parentheses are system global names.
•* Does not apply to Mac SE; alternate sound buffer is not supported.

Source: Inside Macintosh, page IV-257
Technical Introduction to the Macintosh Family, page 141

See Also: 4.09. Macintosh 128/512/Plus/SE Video Buffer
4.10. Macintosh 128/512/Plus/SE Sound Buffer
4.11. Macintosh II System RAM Layout
4.12. Macintosh 128/512/Plus/SE General Memory Organization
4.13. Macintosh II General Memory Organization

4.09. Macintosh 128/512/Plus/SE Video Buffer

Machine	*Memory*	*Main Screen Buffer Location*	*Alternate Screen Buffer Location*
Original 128	128K	$1A700	$12700
512, 512E	512K	$7A700	$72700
Plus/SE	1MB	$FA700	$F2700
Plus/SE	2MB	$1FA700	$1F2700
Plus/SE	2.5MB	$27A700	$272700
Plus/SE	4MB	$3FA700	$3F2700

Notes: • Global variable ScrnBase points to main screen buffer, BufPtr points to alternate screen buffer.
• Buffer is $5580 bytes long (21888 decimal).

Source: Inside Macintosh, pages II-19, IV-247

See Also: 4.08. Macintosh 128/512/Plus/SE System RAM Layout
4.10. Macintosh 128/512/Plus/SE Sound Buffer

4.10. Macintosh 128/512/Plus/SE Sound Buffer

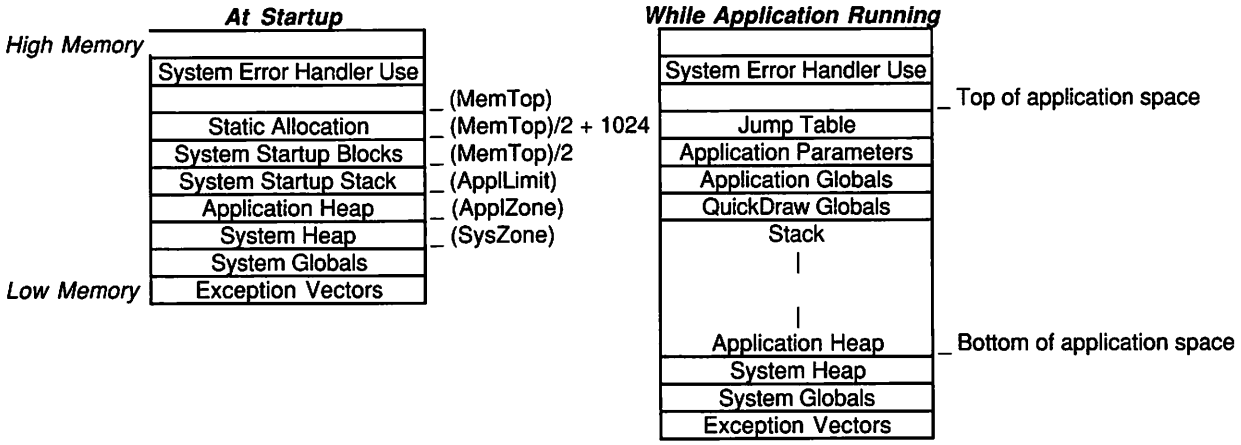
Machine	*Memory*	*Main Sound Buffer Location*	*Alternate Sound Buffer Location*
Original 128	128K	$1FD00	$1A100
512, 512E	512K	$7FD00	$7A100
Plus/SE	1MB	$FFD00	$FA100*
Plus/SE	2MB	$1FFD00	$1FA100*
Plus/SE	2.5MB	$27FD00	$27A100*
Plus/SE	4MB	$3FFD00	$3FA100*

Notes: • Buffer is $2E4 bytes long (740 decimal).
•* Alternate sound buffer does not exist in Mac SE.

Source: Inside Macintosh, pages II-19, IV-247

See Also: 4.08. Macintosh 128/512/Plus/SE System RAM Layout
4.09. Macintosh 128/512/Plus/SE Video Buffer

4.11. Macintosh II System RAM Layout

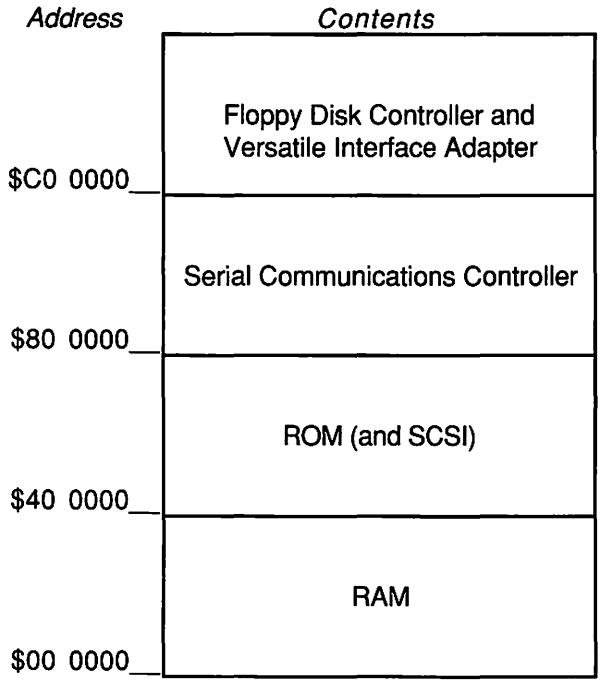


Notes: Names in parentheses are system global names.

Source: Inside Macintosh, page IV-257
Technical Introduction to the Macintosh Family, page 143

See Also: 4.08. Macintosh 128/512/Plus/SE System RAM Layout
4.09. Macintosh 128/512/Plus/SE Video Buffer
4.10. Macintosh 128/512/Plus/SE Sound Buffer
4.12. Macintosh 128/512/Plus/SE General Memory Organization
4.13. Macintosh II General Memory Organization

4.12. Macintosh 128/512/Plus/SE General Memory Organization

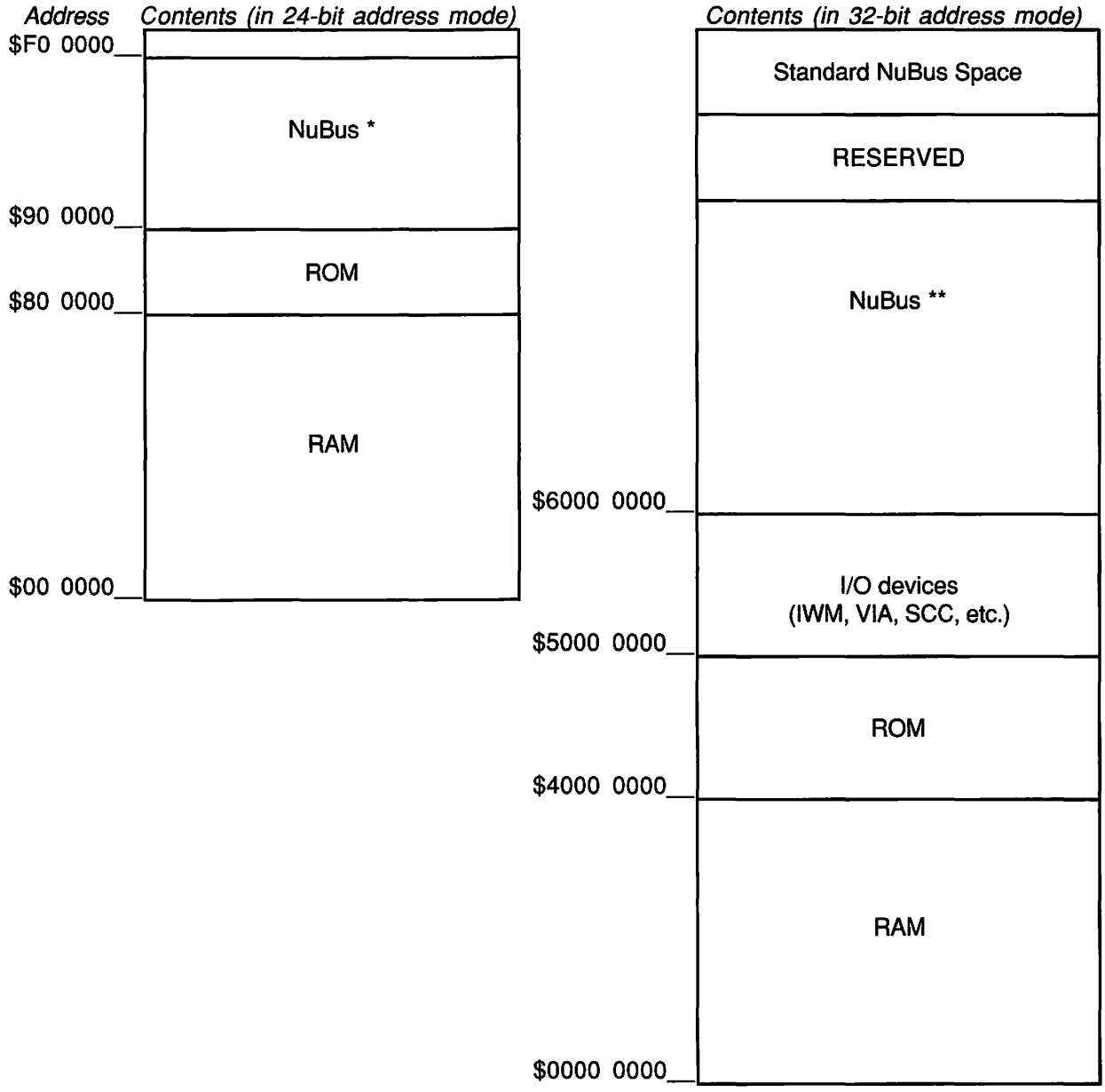


Notes: The entire space devoted to one element is not necessarily used; some items may be repeated, and some blocks of memory left blank.

Source: Technical Introduction to the Macintosh Family, page 204

See Also: 4.08. Macintosh 128/512/Plus/SE System RAM Layout
4.09. Macintosh 128/512/Plus/SE Video Buffer
4.10. Macintosh 128/512/Plus/SE Sound Buffer
4.13. Macintosh II General Memory Organization

4.13. Macintosh II General Memory Organization



Notes:
- * 1 MB assigned to each slot; total of 6 MB assigned to all slots.
- ** 256 MB assigned to each slot; total of 1.5 GB assigned to all slots.

Source: Technical Introduction to the Macintosh Family, pages 204, 205

See Also:
4.08. Macintosh 128/512/Plus/SE System RAM Layout
4.09. Macintosh 128/512/Plus/SE Video Buffer
4.10. Macintosh 128/512/Plus/SE Sound Buffer
4.11. Macintosh II System RAM Layout

4.14. Parameter RAM Layout

Size	*Parameter*	*Values*
Byte	Validity Status	$A8 if data is valid
Byte	Modem Port Node ID Hint	Used by AppleTalk (default = 0)
Byte	Printer Port Node ID Hint	Used by AppleTalk (default = 0)
Byte	Port Use Types	Used by AppleTalk (default = 0)
Int	Modem Port Configuration	See 3.247.
Int	Printer Port Configuration	See 3.247.
Longint	Alarm Setting	Number of seconds since midnight 1 Jan 1904
Int	Application Font	Font ID number minus 1 (default = 2; Geneva)
Int	Autokey Settings	*
Int	Other Settings 1	**
Int	Other Settings 2	***

Notes: •* Consists of four fields formatted as follows:

Bit #s	*Contents*	*Comments*
0	Printer Connect	0 = printer port, 1 = modem port
1–7	RESERVED	
8–11	Autokey rate	In two-tick units
12–15	Autokey threshold	In four-tick units

•** Consists of four fields formatted as follows:

0–3	Caret blink time	In four-tick units
4–7	Doubleclick time	In four-tick units
8–10	Speaker volume	In units, 0 to 7
11–15	RESERVED	

•*** Consists of six fields formatted as follows:

0–1	RESERVED	
2–3	Menu blink setting	Number of blinks, 0 to 3
4	System startup disk	0 = internal, 1 = external
5	RESERVED	
6	Mouse scaling	1 = on, 0 = off
7–15	RESERVED	

Source: Inside Macintosh, pages II-369 to II-373
Technical Introduction to the Macintosh Family, pages 131 to 132

See Also: 3.247. Driver Reset Word

4.15. Real Time Clock Register Commands

Bit Number

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Register*	*Direction*	*Comments*
1	0	0	0	0	0	0	1	Seconds 0	Read	LO byte of seconds
0	0	0	0	0	0	0	1	Seconds 0	Write	LO byte of seconds
1	0	0	0	0	1	0	1	Seconds 1	Read	
0	0	0	0	0	1	0	1	Seconds 1	Write	
1	0	0	0	1	0	0	1	Seconds 2	Read	
0	0	0	0	1	0	0	1	Seconds 2	Write	
1	0	0	0	1	1	0	1	Seconds 3	Read	HO byte of seconds
0	0	0	0	1	1	0	1	Seconds 3	Write	HO byte of seconds
0	0	1	1	0	0	0	1	Test	Write	HO 2 bits of register should be set to 0
0	0	1	1	0	1	0	1	Write Protect	Write	HO bit of register write protects RTC chip
1	0	1	0	x	x	0	1	RAM address 100xx	Read	RAM addresses $10–$13
0	0	1	0	x	x	0	1	RAM address 100xx	Write	RAM addresses $10–$13
1	1	x	x	x	x	0	1	RAM address 0xxxx	Read	RAM addresses $00–$0F
0	1	x	x	x	x	0	1	RAM address 0xxxx	Write	RAM addresses $00–$0F

Notes: LO two bits of the command byte must always be 01 binary.

Source: Inside Macintosh, pages III-37 to III-38

See Also: 4.14. Parameter RAM Layout

4.16. VIA Registers

VIA Register A

Bit Number

7	6	5	4	3	2	1	0	*Register Name*	*Comments*
X								vSCCWReq	SCC wait/request register (1 = char waiting)
	X							vPage2	1 = alternate screen buffer, 0 = main buffer
		X						vHeadSel	Disk select (SEL) line
			X					vOverlay	1 = ROM overlays RAM in low memory
				X				vSndPg2	1 = alternate sound buffer, 0 = main buffer
					X	X	X	vSound (mask)	Sound Volume (0–7)

VIA Register B

X								vSndEnb	1 = sound enabled, 0 = disabled
	X							vH4	1 = video in horiz. blanking period
		X						vY2	Mouse vertical quadrature signal
			X					vX2	Mouse horizontal quadrature signal
				X				vSW	1 = mouse switch pressed, 0 = not pressed
					X			rTCEnb	Real Time Clock Enable
						X		rTCClk	Real Time Clock Clock
							X	rTCData	Real Time Clock Data

VIA Peripheral Control Register

X	X	X							Keyboard Data Interrupt
			X						Keyboard Clock Interrupt
				X	X	X			One-Second Interrupt
							X		Vertical Blanking Interrupt

Source: Inside Macintosh, pages III-39 to III-40

See Also: 4.15. Real Time Clock Register Commands

4.17. VIA Timers

Location	*Timer*	*Comments*
vBase+vT1C	Counter 1	LO byte
vBase+vT1CH	Counter 1	HO byte
vBase+T1L	Latch 1	LO byte
vBase+vT1LH	Latch 1	HO byte
vBase+T2C	Counter 2	LO byte
vBase+vT2CH	Counter 2	HO byte

Notes: You must store the HO and LO bytes separately, because they are not located adjacent in memory.

Source: Inside Macintosh, pages III-40 to III-41

See Also: 4.16. VIA Registers
4.18. VIA Interrupts

4.18. VIA Interrupts

Bit Number

7	6	5	4	3	2	1	0	*Interrupt Name*	*Comments*
X								IRQ	1 = enable INTs, 0 = disable INTs
	X							Timer 1	
		X						Timer 2	
			X					Keyboard Clock	
				X				Keyboard Data Bit	
					X			Keyboard Data Ready	
						X		Vertical Blanking Interrupt	
							X	One-Second Interrupt	

Notes: Bit 7 is always 1 when read.

Source: Inside Macintosh, page III-41

See Also: 4.16. VIA Registers
4.17. VIA Timers

4.19. Exception Vectors

Location	*Vector*	*Comments*
$00	Reset	Not a vector: contains initial stack pointer
$04	Reset	Initial vector
$08	Bus Error	
$0C	Address Error	
$10	Illegal Instruction	
$14	Divide by Zero	
$18	CHK Instruction	
$1C	TRAPV Instruction	
$20	Privilege Violation	
$24	Trace Interrupt	
$28	Line 1010 Emulator	
$2C	Line 1111 Emulator	
$30–$3B	Unassigned	RESERVED
$3C	Uninitialized Interrupt	
$40–$5F	Unassigned	RESERVED
$60	Spurious Interrupt	
$64	VIA Interrupt	
$68	SCC Interrupt	
$6C	VIA+SCC Vector	Temporary
$70	Interrupt Switch	
$74	Interrupt Switch + VIA	
$78	Interrupt Switch + SCC	
$7C	Interrupt Switch + VIA + SCC	
$80–$BF	TRAP Instructions	
$C0–$FF	Unassigned	RESERVED

Source: Inside Macintosh, pages III-45 to III-46

See Also: 4.26. 680x0 Exception Vectors

4.20. SCC Data and Control Registers

Location	*Register*	*Comments*
sccWBase+aData	Data A	Write
sccRBase+aData	Data A	Read
sccWBase+bData	Data B	Write
sccRBase+bData	Data B	Read
sccWBase+aCtl	Control A	Write
sccRBase+aCtl	Control A	Read
sccWBase+bCtl	Control B	Write
sccRBase+bCtl	Control B	Read

Notes: • Use only even-address reads, odd-address writes.
• SCC chip should not be accessed more often than every 2.2 microseconds.

Source: Inside Macintosh, page III-25

4.21. SCSI Address Assignments

Address	*Device*	*Comments*
0		Lowest priority boot device
1		
2	CMS drives	
3		
4		
5	MacBottom drives	
6		Highest priority boot device
7	Macintosh	

Notes: No two SCSI devices should occupy the same address.

Source: CMS Disk Manual, page 26

4.22. Disk State Control Lines and Registers

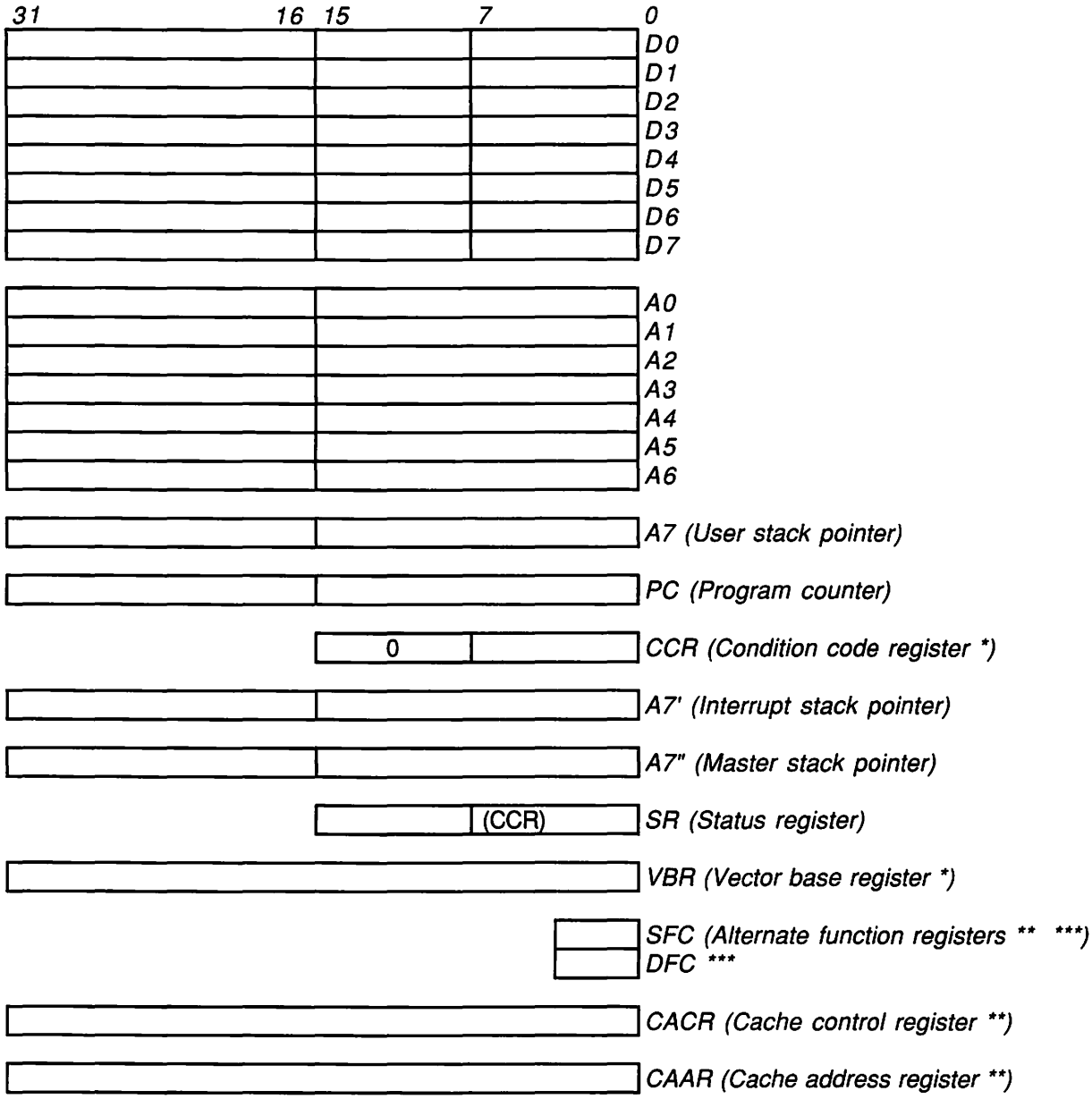
State Control Lines

CA2	*CA1*	*CA0*	*SEL*	*Register*	*Direction*	*Comments*
0	0	0	0	DIRTN	Read/write	Direction of head step (0 = in, 1 = out)
0	0	0	1	CSTIN	Read only	Disk in place (0 = disk in drive)
0	0	1	0	STEP	Read/write	Disk head step (0 = step one track)
0	0	1	1	WRTPRT	Read only	Disk locked (0 = disk is locked)
0	1	0	0	MOTORON	Read/write	Disk motor running (0 = motor on, 1 = motor off)
0	1	1	0	EJECT	Write only	Disk eject (1 = eject)
0	1	0	1	TKO	Read only	Head is at track zero (0 = head is at track 0)
0	1	1	1	TACH	Read only	Tachometer (pulse train indicates disk speed)
1	0	0	0	RDDATA0	Read only	Read data from lower head
1	0	0	1	RDDATA1	Read only	Read data from upper head
1	1	0	0	SIDES	Read only	Single- or double-sided drive (0 = single-sided, 1 = double-sided)
1	1	1	1	DRVIN	Read only	Drive installed (0 = drive connected, 1 = no drive installed)

Notes: See source for directions on how to trigger read/write states.

Source: Inside Macintosh, pages III-34 to III-36

4.23. 680x0 Register Set



Notes:
- * 68020 extends register to 16 bits; 68000 register is 8 bits.
- ** 68020 only.
- *** Only LO three bits actually used.

Source: M68000 Programmer's Reference Manual, page 1-2
MC68020 32-bit Microprocessor User's Manual, page 1-3

See Also: 1.10. Common 680x0 Data Formats
4.24. 680x0 Instruction Set
4.25. 680x0 Addressing Modes
4.43. 68000 Pinouts
4.44. 68020 Pinouts

4.24. 680x0 Instruction Set

Mnemonic	*Instruction Description*	*Syntax*
ABCD	Add decimal with extend	ABCD -(Ay), -(Ax)
ADD	Add	ADD Dn,<ea>
ADDA	Add address	ADDA <ea>,An
ADDI	Add immediate	ADDI #<data>,<ea>
ADDQ	Add quick	ADDQ #<data>,<ea>
ADDX	Add with extend	ADDX -(Ay),-(Ax)
AND	Logical AND	AND Dn,<ea>
ANDI	Logical AND immediate	ANDI #<data>,<ea> ANDI #<data>,CCR ANDI #<data>,SR ***
ASL	Arithmetic shift left	ASL #<data>,Dy ASL <ea>
ASR	Arithmetic shift right	ASR #<data>,Dy ASR <ea>
Bcc	Branch conditionally	Bcc <label>
BCHG	Test bit and change	BCHG Dn,<ea> BCHG #<data>,<ea>
BCLR	Test bit and clear	BCLR Dn,<ea> BCLR #<data>,<ea>
BFCHG *	Test bit field and change	BFCHG <ea> {offset:width}
BFCLR *	Test bit field and clear	BFCLR <ea> {offset:width}
BFEXTS *	Bit field extract (signed)	BFEXTS <ea> {offset:width},Dn
BFEXTU *	Bit field extract (unsigned)	BFEXTU <ea> {offset:width},Dn
BFFFO *	Bit field find first one	BFFFO <ea> {offset:width},Dn
BFINS *	Bit field insert	BFINS Dn,<ea> {offset:width}
BFSET *	Test bit field and set	BFSET <ea> {offset:width}
BFTST *	Test bit field	BFTST <ea> {offset:width}
BKPT	Breakpoint	BKPT #<data>
BRA	Branch	BRA <label>
BSET	Test bit and set	BSET Dn,<ea> BSET #<data>,<ea>
BSR	Branch to subroutine	BSR <label>
BTST	Test bit	BTST Dn,<ea> BTST #<data>,<ea>
CALLM *	Call module	CALLM #<data>,<ea>
CAS *	Compare and swap operands	CAS Dc,Du,<ea>
CAS2 *	Compare and swap dual operands	CAS2 Dc1:Dc2,Du1:Du2,(Rn1):(Rn2)
CHK	Check register against bound	CHK <ea>,Dn
CHK2 *	Check register against upper and lower bounds	CHK2 <ea>,Rn
CLR	Clear	CLR <ea>
CMP	Compare	CMP <ea>,Dn
CMPA	Compare address	CMPA <ea>,An
CMPI	Compare immediate	CMPI #<data>,<ea>
CMPM	Compare memory to memory	CMPM (Ay) +, (Ax) +
CMP2 *	Compare register against upper and lower bounds	CMP2 <ea>,Rn
DBcc	Test condition decrement and branch	DBcc Dn,<label>
DIVS * *	Signed divide	DIVS.W <ea>,Dn DIVS.L <ea>,Dq DIVS.L <ea>,Dr,:Dq
DIVSL *	Signed divide	DIVSL.L <ea>,Dr:Dq
DIVU * *	Unsigned divide	DIVS.W <ea>,Dn DIVU.L <ea>,Dq DIVU.L <ea>,Dr:Dq
DIVUL *	Unsigned divide	DIVUL.L <ea>,Dr:Dq
EOR	Logical exclusive OR	EOR Dn,<ea>
EORI	Logical exclusive OR immediate	EORI #<data>,<ea> EORI #<data>,CCR EORI #<data>,SR ***

(continued)

Table 4.24. Continued

Mnemonic	*Instruction Description*	*Syntax*
EXG	Exchange registers	EXG Dx,Dy EXG Ax,Ay EXG Dx,Ay
EXT	Sign extend	EXT.W Dn EXT.L Dn
EXTB *	Sign extend	EXTB.L Dn
ILLEGAL	Take illegal instruction trap	ILLEGAL
JMP	Jump	JMP <ea>
JSR	Jump to subroutine	JSR <ea>
LEA	Load effective address	LEA <ea>,An
LINK	Link and allocate	LINK An,#<displacement>
LSL	Logical shift left	LSL Dx,Dy LSL #<data>,Dy LSL <ea>
LSR	Logical shift right	LSR Dx,Dy LSR #<data>,Dy LSR <ea>

MOVE	Move	MOVE <ea>,<ea>
		MOVE <ea>,CCR
		MOVE <ea>,SR ***

MOVEA	Move address	MOVEA <ea>,An
MOVE CCR	Move condition code register	MOVECCR, <ea>
MOVE SR	Move status register	MOVESR,<ea>
MOVE USP ***	Move user stack pointer	MOVE USP,An MOVE An,USP
MOVEC	Move control register	MOVEC Rc,Rn MOVEC Rn,Rc
MOVEM	Move multiple registers	MOVEM registerlist,<ea> MOVEM <ea>,registerlist
MOVEP	Move peripheral	MOVEP Dx,(d,Ay) MOVEP (d,Ay)Dx
MOVEQ	Move quick	MOVEQ #<data>,Dn
MOVES	Move alternate address space	MOVES Rn,<ea> MOVES <ea>,Rn
MULS	Signed multiply	MULS.W <ea>,Dn MULS.L <ea>,Dl * MULS.L <ea>,Dh:Dl *
MULU	Unsigned multiply	MULU.W <ea>,Dn MULU.L <ea>,Dl * MULU.L <ea>,Dh:Dl *
NBCD	Negate decimal with extend	NBCD <ea>
NEG	Negate	NEG <ea>
NEGX	Negate with extend	NEGX <ea>
NOP	No operation	NOP
NOT	Logical complement	NOT <ea>
OR	Logical inclusive OR	OR <ea>,Dn OR Dn,<ea>
ORI	Logical inclusive OR immediate	ORI #<data>,<ea> ORI #<data>,CCR ORI #<data>,SR ***
PACK *	Pack BCD	PACK -(Ax),-(Ay),#<adjustment> PACK Dx,Dy,#<adjustment>
PEA	Push effective address	PEA <ea>
RESET	Reset external devices	RESET ***
ROL	Rotate left	ROL Dx,Dy ROL #<data>,Dy ROL <ea>
ROR	Rotate right	ROR Dx,Dy ROR #<data>,Dy ROR <ea>
ROXL	Rotate left with extend	ROXL Dx,Dy ROXL #<data>,Dy ROXL <ea>

(continued)

Table 4.24. Continued

Mnemonic	*Instruction Description*	*Syntax*
ROXR	Rotate right with extend	ROXR Dx,Dy ROXR #<data>,Dy ROXR <ea>
RTD	Return and deallocate	RTD #<displacement>
RTE ***	Return from exception	RTE
RTM *	Return from module	RTM Rn
RTR	Return and restore codes	RTR
RTS	Return from subroutine	RTS
SBCD	Subtract decimal with extend	SBCD Dx,Dy SBCD -(Ax),-(Ay)
Scc	Set conditionally	Scc <ea>
STOP	Stop	STOP #<data>
SUB	Subtract	SUB <ea>,Dn SUB Dn,<ea>
SUBA	Subtract address	SUBA <ea>,An
SUBI	Subtract immediate	SUBI #<data>,<ea>
SUBQ	Subtract quick	SUBQ #<data>,<ea>
SUBX	Subtract with extend	SUBX Dx,Dy SUBX -(Ax), -(Ay)
SWAP	Swap register halves	SWAP Dn
TAS	Test operand and set	TAS <ea>
TRAP	Trap	TRAP #<vector>
TRAPcc *	Trap conditionally	TRAPcc TRAPcc.W #<data> TRAPcc.L #<data>
TRAPV	Trap on overflow	TRAPV
TST	Test operand	TST <ea>
UNLK	Unlink	UNLK An
UNPK *	Unpack BCD	UNPACK -(Ax), -(Ay),#<adjustment>
cpBcc **	Branch conditionally	cpBcc <label>
cpDBcc **	Test coprocessor condition decrement and branch	cpDBcc Dn,<label>
cpGEN **	Coprocessor general instruction	cpGEN <parameters>
cpRESTORE **	Restore internal state of coprocessor ***	cpRESTORE <ea>
cpSAVE **	Save internal state of coprocessor ***	cpSAVE <ea>
cpScc **	Set conditionally	cpScc <ea>
cpTRAPcc **	Trap conditionally	cpTRAPcc cpTRAPcc #<data>

Notes:
- * 68020 only.
- ** Coprocessor instruction.
- *** Privileged instruction.

Legend:
Dn = data register direct (n = register number)
An = address register direct (n = register number)
(An) = address register indirect
(An)+ = address register indirect with postincrement
-(An) = address register indirect with predecrement
.W = short
.L = long
#<data> = immediate data
<ea> = effective address
Other abbreviations in Syntax field follow Motorola conventions.

Source:
M68000 Programmer's Reference Manual, pages 1-5, B1 to B116
MC68020 32-bit Microprocessor User's Manual, pages 1-6, B1 to B170

See Also:
1.10. Common 680x0 Data Formats
4.23. 680x0 Register Set
4.25. 680x0 Addressing Modes

4.25. 680x0 Addressing Modes

Mode	*Effective Address*	*Other*
Data Register Direct	Dn	
Address Register Direct	An	
Absolute Short	(next word)	
Absolute Long	(next doubleword)	
PC Relative with Offset	(PC) + d16	
PC Relative with Index and Offset	(PC) + d8	
PC Relative Memory Indirect Post-Indexed *	(bd + PC) + Xn.SIZE*SCALE + od	
PC Relative Memory Indirect Pre-Indexed *	(bd + PC + Xn.SIZE*SCALE) + od	
Register Indirect	(An)	
Postincrement Register Indirect	(An)	An <- An + N (after calculation)
Predecrement Register Indirect	(An)	An <- An - N (before calculation)
Register Indirect with Offset	(An) + d16	
Indexed Register Indirect with Offset	(An) + (Xn) + d8	
Indexed Register with Base Offset *	(An) + (Xn) + bd	
Memory Indirect Post-Indexed *	(bd + An) + Xn.SIZE*SCALE + od	
Memory Indirect Pre-Indexed *	(bd + An + Xn.SIZE*SCALE) + od	
Immediate		Data = next word(s)
Quick Immediate		Inherent data
Implied Register	SR, USP, SSP, PC, VBR, SFC, or DFC	

Notes: * 68020 only.

Legend:
Dn = data register direct (n = register number)
An = address register direct (n = register number)
(An) = address register indirect
(An)+ = address register indirect with postincrement
-(An) = address register indirect with predecrement
Other abbreviations in Syntax field follow Motorola conventions.

Source:
M68000 Programmer's Reference Manual, page1-4
MC68020 32-bit Microprocessor User's Manual, pages 2-6 to 2-19

See Also:
1.10. Common 680x0 Data Formats
4.23. 680x0 Register Set

4.26. 680x0 Exception Vectors

Vector #	*Address*	*Vector Assignment*
0	$00	Reset: initial SSP
1	$04	Reset: initial PC
2	$08	Bus error
3	$0C	Address error
4	$10	Illegal instruction
5	$14	Zero divide
6	$18	CHK instruction (CHK2 also on 68020)
7	$1C	TRAPV, cpTRAPcc, TRAPcc instructions
8	$20	Privilege violation
9	$24	Trace
10	$28	Line 1010 emulator
11	$2C	Line 1111 emulator
12	$30	RESERVED
13 *	$34	Coprocessor protocol violation
14	$38	Format error
15	$3C	Uninitialized interrupt vector
16–23	$40–$5C	RESERVED
24	$60	Spurious interrupt
25	$64	Level 1 interrupt autovector
26	$68	Level 2 interrupt autovector
27	$6C	Level 3 interrupt autovector
28	$70	Level 4 interrupt autovector
29	$74	Level 5 interrupt autovector
30	$78	Level 6 interrupt autovector
31	$7C	Level 7 interrupt autovector
32–47	$80–$BC	TRAP instruction vectors
48 *	$C0	FPCP branch or set on unordered condition
49 *	$C4	FPCP inexact result
50 *	$C8	FPCP divide by zero
51 *	$CC	FPCP underflow
52 *	$D0	FPCP operand error
53 *	$D4	FPCP overflow
54 *	$D8	FPCP signaling NAN
55 *	$DC	RESERVED
56 *	$E0	PMMU configuration
57 *	$E4	PMMU illegal operation
58 *	$E8	PMMU access level violation
59–63	$EC–$FF	RESERVED
64–255	$100–$3FC	User-defined

Notes: * Defined in the 68020 reference only.

Source: M68000 Programmer's Reference Manual, page 4-5
MC68020 32-bit Microprocessor User's Manual, pages 6-5 to 6-6

See Also: 4.19. Exception Vectors

4.27. 680x0 Address Space Types

Function Code Output

FC2	*FC1*	*FC0*	*Address Space*
0	0	0	RESERVED
0	0	1	User data space
0	1	0	User program space
0	1	1	User-defined
1	0	0	RESERVED
1	0	1	Supervisor data space
1	1	0	Supervisor program space
1	1	1	CPU space (interrupt acknowledge)

Source: M68000 Programmer's Reference Manual, page 4-3
MC68020 32-bit Microprocessor User's Manual, page 6-4

4.28. 680x0 Normal Stack Frames

15 *0*

Status Register		<-----*SP*
Program Counter (High)		*+$02*
Program Counter (Low)		*+$04*
0 0 0 0	Vector Offset	*+$06**

Status Register		<-----*SP**
Program Register (High)		*+$02**
Program Register (Low)		*+$04**
0 0 0 1	Vector Offset	*+$06**

Status Register		<-----*SP*
Program Counter (High)		*+$02**
Program Counter (Low)		*+$04**
0 0 1 0	Vector Offset	*+$06**
Instruction Address (High)		*+$08**
Instruction Address (Low)		*+$0A**

Notes: * Not used on 68000.

Source: M68000 Programmer's Reference Manual, page 4-7
MC68020 32-bit Microprocessor User's Manual, pages 6-20 to 6-21

See Also: 4.29. 680x0 Exception Stack Frames
4.30. 68020 Long/Short Bus Cycle Fault Stack Frames
4.31. 68020 Module Call Stack Frame

4.29. 680x0 Exception Stack Frames

68000 Exception Stack (Bus and Address Error)

15		0	
	Status Register		<-----SP
	Program Counter (High)		+$02
	Program Counter (Low)		+$04
1 0 0 0	Vector Offset		+$06
	Special Status Word		+$08
	Fault Address (High)		+$0A
	Fault Address (Low)		+$0C
	RESERVED		+$0E
	Data Output Buffer		+$10
	RESERVED		+$12
	Data Input Buffer		+$14
	RESERVED		+$16
	Instruction Input Buffer		+$18
	16 Words of Internal Info		+$1A

68020 Exception Stack (Coprocessor Mid-Instruction)

15		0	
	Status Register		<-----SP
	Program Counter (High)		+$02
	Program Counter (Low)		+$04
1 0 0 1	Vector Offset		+$06
	Instruction Address (High)		+$08
	Instruction Address (Low)		+$0A
	4 Words of Internal Registers		+$0C

Source: M68000 Programmer's Reference Manual, page 4-14
MC68020 32-bit Microprocessor User's Manual, page 6-21

See Also: 4.28. 680x0 Normal Stack Frames
4.30. 68020 Long/Short Bus Cycle Fault Stack Frames
4.31. 68020 Module Call Stack Frame

4.30. 68020 Long/Short Bus Cycle Fault Stack Frames

68020 Short Bus Cycle Fault Stack Frame

15	0	
Status Register		<-----SP
Program Counter (High)		+$02
Program Counter (Low)		+$04
1 0 1 0	Vector Offset	+$06
Internal Register		+$08
Special Status Word		+$0A
Instruction Pipe Stage C		+$0C
Instruction Pipe Stage B		+$0E
Data Cycle Fault Address (High)		+$10
Data Cycle Fault Address (Low)		+$12
Internal Register		+$14
Internal Register		+$16
Data Output Buffer (High)		+$18
Data Output Buffer (Low)		+$1A
Internal Register		+$1C
Internal Register		+$1E

68020 Long Bus Cycle Fault Stack Frame

15	0	
Status Register		<-----SP
Program Counter (High)		+$02
Program Counter (Low)		+$04
1 0 1 1	Vector Offset	+$06
Internal Register		+$08
Special Status Word		+$0A
Instruction Pipe Stage C		+$0C
Instruction Pipe Stage B		+$0E
Data Cycle Fault Address (High)		+$10
Data Cycle Fault Address (Low)		+$12
Internal Register		+$14
Internal Register		+$16
Data Output Buffer (High)		+$18
Data Output Buffer (Low)		+$1A
Internal Register		+$1C
Internal Register		+$1E
Internal Register		+$20
Internal Register		+$22
Stage B Address (High)		+$24
Stage B Address (Low)		+$26
Internal Register		+$28
Internal Register		+$2A
Data Input Buffer (High)		+$2C
Data Input Buffer (Low)		+$2E
22 Words Internal Registers		+$30

Source: MC68020 32-bit Microprocessor User's Manual, pages 6-22 to 6-23

See Also: 4.28. 680x0 Normal Stack Frames
4.29. 680x0 Exception Stack Frames
4.31. 68020 Module Call Stack Frame

4.31. 68020 Module Call Stack Frame

15	*12*	*8*	*0*	
Opt	Type	Saved Access Level		<-----*SP*
0 0 0	0 0 0 0 0	Condition Codes		*+$02*
0 0 0	0 0 0 0 0	Argument Codes		*+$04*
RESERVED				*+$06*
Module Description Pointer (High)				*+$08*
Module Description Pointer (Low)				*+$0A*
Saved Program Counter (High)				*+$0C*
Saved Program Counter (Low)				*+$0E*
Saved Module Data Area Pointer (High)				*+$10*
Saved Module Data Area Pointer (Low)				*+$12*
Saved Stack Pointer (High)				*+$14*
Saved Stack Pointer (Low)				*+$16*
Optional Arguments				*+$18*

Source: MC68020 32-bit Microprocessor User's Manual, page D-3

See Also: 4.28. 680x0 Normal Stack Frames
4.29. 680x0 Exception Stack Frames
4.30. 68020 Long/Short Bus Cycle Fault Stack Frames

4.32. 680x0 Special Status Word

68000

Bit #	*Function*	*Description*
0	FC0	Function code 0
1	FC1	Function code 1
2	FC2	Function code 2
3	-	RESERVED
4	-	RESERVED
5	-	RESERVED
6	-	RESERVED
7	-	RESERVED
8	RW	Read/write flag (0 = write,1 = read)
9	BY	Byte transfer flag
10	HB	High byte transfer
11	RM	Read-modify-write cycle
12	DF	Data fetch to input buffer
13	IF	Instruction fetch to input buffer
14	-	RESERVED
15	RR	Rerun flag (0 = proc rerun, 1 = software rerun)

(continued)

Table 4.32. Continued

68020

Bit #	*Function*	*Description*
0	FC0	Function code 0
1	FC1	Function code 1
2	FC2	Function code 2
3	0	
4	SIZ	HO size code for data cycle
5	SIZ	LO size code for data cycle
6	RW	Read/write flag (0 = write, 1 = read)
7	RM	Read-modify-write cycle
8	DF	Fault rerun flag for data cycle (0 = do not rerun)
9	0	
10	0	
11	0	
12	RB	Rerun flag for stage B of instruction pipe (0 = do not rerun)
13	RC	Rerun flag for stage C of instruction pipe (0 = do not rerun)
14	FB	Fault on stage B of instruction pipe
15	FC	Fault on stage C of instruction pipe

Source: M68000 Programmer's Reference Manual, page 4-15
MC68020 32-bit Microprocessor User's Manual, page 6-17

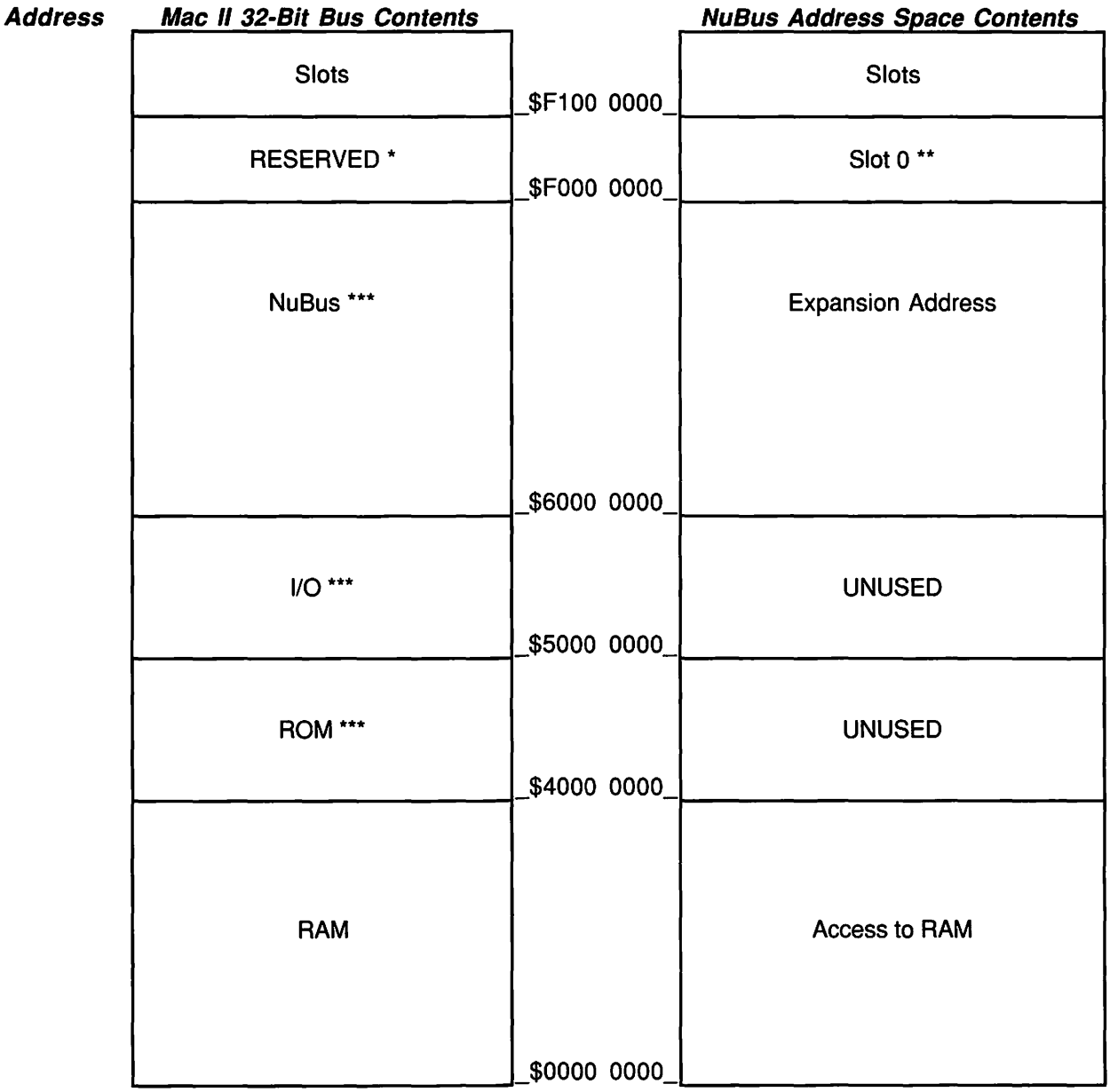
4.33. 680x0 Coprocessor Interface Register Set Map

31 — *16*	*15* — *0*	
Response	Control	*$00*
Save	Restore	*$04*
Operation Word	Command	*$08*
RESERVED	Condition	*$0C*
Operand		*$10*
Register Select	RESERVED	*$14*
Instruction Address		*$18*
Operand Address		*$1C*

Source: MC68020 32-bit Microprocessor User's Manual, pages 8-5 to 8-6

See Also: 4.23. 680x0 Register Set

4.34. NuBus Memory Usage



Notes:
- * Used by Macintosh system; will generate bus error exception if referenced.
- ** NuBus $F080 0000 to $F0FF FFFF is remapped to Apple Bus $4080 0000 to 40FF FFFF. NuBus $F000 0000 to $F070 FFFF is remapped to Apple Bus $5000 0000.
- *** $6000 0000 to $8FFF FFFF presently unused; $9000 0000 up for super slots $9 to $E.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 1-10, 4-5

See Also: 4.13. Macintosh II General Memory Organization
4.35. NuBus Slot Memory Assignments
4.37. NuBus to 68020 Byte Swapping

4.35. NuBus Slot Memory Assignments

NuBus Slot	*Macintosh 24-Bit Addr.*	*NuBus Super Slot Addr.*	*NuBus Slot Space*
$9	$90 0000 to $9F FFFF	$9000 0000 to $9FFF FFFF	$F900 0000 to $F9FF FFFF
$A	$A0 0000 to $AF FFFF	$A000 0000 to $AFFF FFFF	$FA00 0000 to $FAFF FFFF
$B	$B0 0000 to $BF FFFF	$B000 0000 to $BFFF FFFF	$FB00 0000 to $FBFF FFFF
$C	$C0 0000 to $CF FFFF	$C000 0000 to $CFFF FFFF	$FC00 0000 to $FCFF FFFF
$D	$D0 0000 to $DF FFFF	$D000 0000 to $DFFF FFFF	$FD00 0000 to $FDFF FFFF
$E	$E0 0000 to $EF FFFF	$E000 0000 to $EFFF FFFF	$FE00 0000 to $FEFF FFFF

Notes: Only one megabyte of NuBus card space is mapped into the 24-bit Macintosh addressing mode—the remainder must be addressed in 32-bit mode.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 4-6

See Also: 4.34. NuBus Memory Usage
4.36. NuBus 24-Bit to 32-Bit Memory Translations

4.36. NuBus 24-Bit to 32-Bit Memory Translations

Macintosh 24-Bit Address	*NuBus 32-Bit Address*
$00 0000 to $7F FFFF	$0000 0000 to $007F FFFF
$80 0000 to $8F FFFF	$4000 0000 to $400F FFFF
$#0 0000 to $#F FFFF *	$F#00 0000 to $F#0F FFFF *
$F0 0000 to $FF FFFF	$5000 0000 to $500F FFFF

Notes: * # Refers to NuBus Super Slot number ($9 to $E).

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 4-4

See Also: 4.34. NuBus Memory Usage
4.35. NuBus Slot Memory Assignments

4.37. NuBus to 68020 Byte Swapping

NuBus Bits	*68020 Bits*
0 to 7 (LSB)	24 to 31 (MSB)
8 to 15	16 to 23
16 to 23	8 to 15
24 to 31 (MSB)	0 to 7 (LSB)

Notes: Bits are mapped a byte at a time. Thus bit 2 on the NuBus corresponds to bit 26 on the 68020.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 4-7 to 4-8

See Also: 4.38. NuBus ByteLanes Mapping

4.38. NuBus ByteLanes Mapping

*ByteLanes Value (in Bits) **

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*ByteLanes Value*	*ByteLanes Address **
1	1	1	0	0	0	0	1	$E1	$F#FF FFFC
1	1	0	1	0	0	1	0	$D2	$F#FF FFFD
1	1	0	0	0	0	1	1	$C3	$F#FF FFFD
1	0	1	1	0	1	0	0	$B4	$F#FF FFFE
1	0	1	0	0	1	0	1	$A5	$F#FF FFFE
1	0	0	1	0	1	1	0	$96	$F#FF FFFE
1	0	0	0	0	1	1	1	$87	$F#FF FFFE
0	1	1	1	1	0	0	0	$78	$F#FF FFFF
0	1	1	0	1	0	0	1	$69	$F#FF FFFF
0	1	0	1	1	0	1	0	$5A	$F#FF FFFF
0	1	0	0	1	0	1	1	$4B	$F#FF FFFF
0	0	1	1	1	1	0	0	$3C	$F#FF FFFF
0	0	1	0	1	1	0	1	$1E	$F#FF FFFF
0	0	0	0	1	1	1	1	$0F	$F#FF FFFF

Notes: •* LO nibble is set for each byte lane (bits 0–3); HO nibble is complement.
•** # represents NuBus slot number.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 8-5 to 8-6

See Also: 4.37. NuBus to 68020 Byte Swapping

4.39. NuBus Block Transfer Size and Address

CPU Signals

/AD5	*/AD4*	*/AD3*	*/AD2*	*Block Size*	*Starting Address of Block*
-	-	-	1	2 words	(AD31-AD3)000
-	-	1	0	4 words	(AD31-AD4)0000
-	1	0	0	8 words	(AD31-AD5)00000
1	0	0	0	16 words	(AD31-AD6)000000
0	0	0	0	Error	NA

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 3-12

4.40. TMx Transfer Mode Codes

CPU Signals

/TM1	*/TM0*	*/AD1*	*/AD0*	*Direction*	*Size **
0	0	0	0	Write	Byte 3
0	0	0	1	Write	Byte 2
0	0	1	0	Write	Byte 1
0	0	1	1	Write	Byte 0
0	1	0	0	Write	Halfword 1
0	1	0	1	Write	Block
0	1	1	0	Write	Halfword 0
0	1	1	1	Write	Word
1	0	0	0	Read	Byte 3
1	0	0	1	Read	Byte 2
1	0	1	0	Read	Byte 1
1	0	1	1	Read	Byte 0
1	1	0	0	Read	Halfword 1
1	1	0	1	Read	Block
1	1	1	0	Read	Halfword 0
1	1	1	1	Read	Word

Notes: * Byte 0 is bits 0–7, byte 1 is bits 8–15, byte 2 is bits 16–23, byte 3 is bits 24–31, halfword 0 is bits 0–15, halfword 1 is bits 16–31.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 3-5, 3-6

See Also: 4.41. TMx Acknowledgment Codes
4.42. TMx Attention Cycle Codes

4.41. TMx Acknowledgment Codes

CPU Signals

/TM1	*/TM0*	*Acknowledgment Type*
0	0	Bus Transfer Complete
0	1	Error
1	0	Bus Timeout Error
1	1	Try Again Later

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 3-9

See Also: 4.40. TMx Transfer Mode Codes
4.42. TMx Attention Cycle Codes

4.42. TMx Attention Cycle Codes

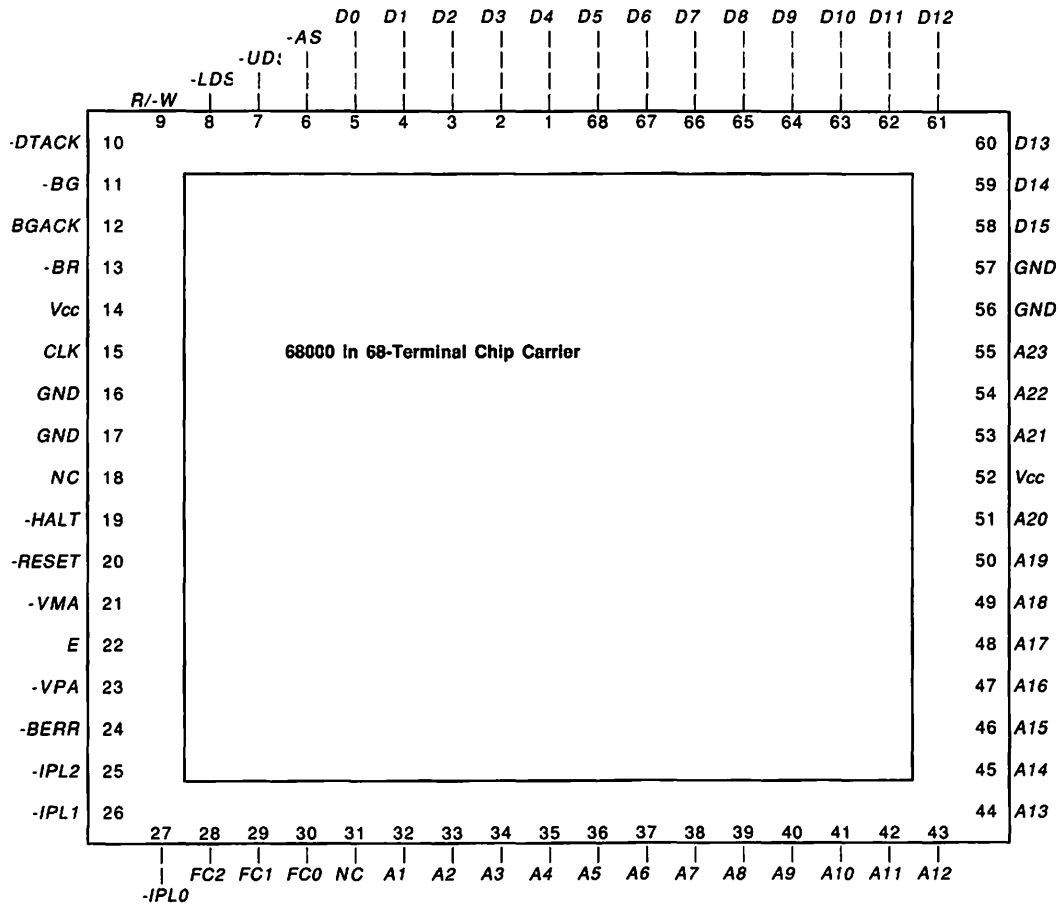
CPU Signals

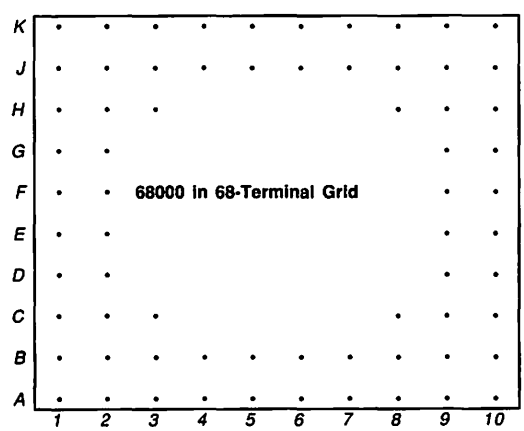
/TM1	/TM0	Attention Cycle Type
0	0	Attention-Null
0	1	RESERVED
1	0	Attention-Resource Lock
1	1	RESERVED

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 3-10

See Also: 4.40. TMx Transfer Mode Codes
4.41. TMx Acknowledgment Codes

4.43. 68000 Pinouts





(continued)

Table 4.43. Continued

Pin	Signal
A1	NC
A2	-AS
A3	D1
A4	D2
A5	D4
A6	D5
A7	D7
A8	D8
A9	D10
A10	D12

Pin	Signal
B1	-DTACK
B2	-LDS
B3	-UDS
B4	D0
B5	D3
B6	D6
B7	D9
B8	D11
B9	D13
B10	D15

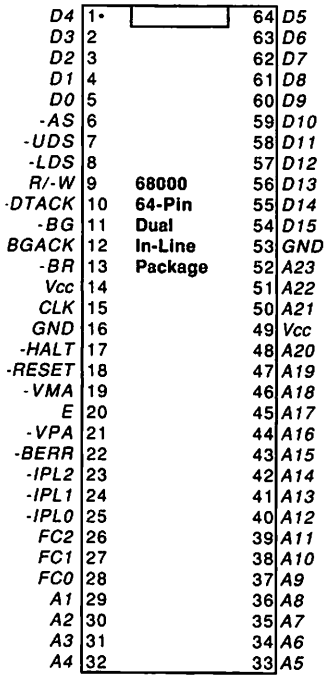
Pin	Signal
C1	-BGACK
C2	-BG
C3	R/-W
C8	D13
C9	A23
C10	A22
D1	-BR
D2	Vcc
D9	Vss
D10	A21

Pin	Signal
E1	CLK
E2	Vss
E9	Vcc
E10	A20
F1	-HALT
F2	-RESET
F9	A18
F10	A19
G1	-VMA
G2	-VPA

Pin	Signal
G9	A15
G10	A17
H1	E
H2	-IPL2
H3	-IPL1
H8	A13
H9	A12
H10	A16
J1	-BERR
J2	-IPL0

Pin	Signal
J3	FC1
J4	NC
J5	A2
J6	A5
J7	A8
J8	A10
J9	A11
J10	A14
K1	NC
K2	FC2

Pin	Signal
K3	FC0
K4	A1
K5	A3
K6	A4
K7	A6
K8	A7
K9	A9
K10	NC

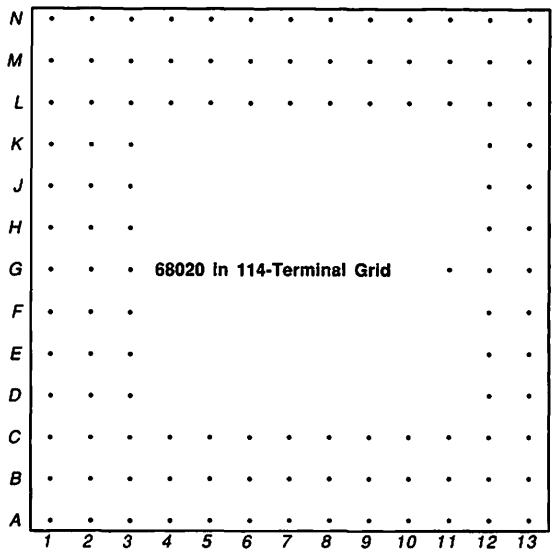


Notes: Minus sign (-) in front of signal name indicates negative going signal.

Source: 68000 Microprocessor Handbook, pages 131 to 134

See Also: 4.44. 68020 Pinouts
4.45. 68881 Pinouts
4.46. 68851 Pinouts

4.44. 68020 Pinouts



Pin	Signal
A1	-BGACK
A2	A1
A3	A31
A4	A28
A5	A26
A6	A23
A7	A22
A8	A19
A9	Vcc
A10	GND
A11	A14
A12	A11
A13	A8

Pin	Signal
B1	NC
B2	-BG
B3	-BR
B4	A30
B5	A27
B6	A24
B7	A20
B8	A18
B9	GND
B10	A15
B11	A13
B12	A10
B13	A6

Pin	Signal
C1	-RESET
C2	CLOCK
C3	NC
C4	A0
C5	A29
C6	A25
C7	A21
C8	A17
C9	A16
C10	A12
C11	A9
C12	A7
C13	A5

Pin	Signal
D1	Vcc
D2	Vcc
D3	NC
D12	A4
D13	A3
E1	FC0
E2	-RMC
E3	Vcc
E12	A2
E13	-OCS
F1	SIZ0
F2	FC2
F3	FC1

Pin	Signal
F12	NC
F13	-IPEND
G1	-ECS
G2	SIZ1
G3	-DBEN
G11	Vcc
G12	GND
G13	Vcc
H1	-CDIS
H2	-AVEC
H3	-DSACK0
H12	IPL2
H13	GND

Pin	Signal
J1	-DSACK1
J2	-BERR
J3	GND
J12	-IPL0
J13	-IPL1
K1	GND
K2	-HALT
K3	NC
K12	D1
K13	D0

Pin	Signal
L1	-AS
L2	R/-W
L3	D30
L4	D27
L5	D23
L6	D19
L7	GND
L8	D15
L9	D11
L10	D7
L11	NC
L12	D3
L13	D2

Pin	Signal
M1	-DS
M2	D29
M3	D26
M4	D24
M5	D21
M6	D18
M7	D16
M8	Vcc
M9	D13
M10	D10
M11	D6
M12	D5
M13	D4

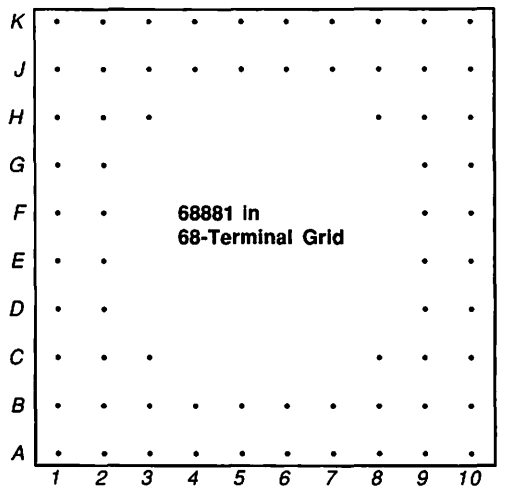
Pin	Signal
N1	D31
N2	D28
N3	D25
N4	D22
N5	D20
N6	D17
N7	GND
N8	Vcc
N9	D14
N10	D12
N11	D9
N12	D8
N13	NC

Notes: Minus sign (-) in front of signal name indicates negative going signal.

Source: 68000 Microprocessor Handbook, pages 131 to 133

See Also: 4.43. 68000 Pinouts
4.45. 68881 Pinouts
4.46. 68851 Pinouts

4.45. 68881 Pinouts



Pin	Signal
A1	Vcc (Reserved)
A2	GND
A3	D0
A4	D1
A5	D3
A6	D4
A7	D6
A8	D7
A9	D8
A10	GND (Reserved)

Pin	Signal
B1	Vcc
B2	GND
B3	GND
B4	-SENSE
B5	D2
B6	D5
B7	GND
B8	Vcc
B9	D10
B10	D11

Pin	Signal
C1	GND
C2	CLK
C3	GND
C8	D9
C9	D13
C10	D14
D1	-RESET
D2	GND (Reserved)
D9	D12
D10	D15

Pin	Signal
E1	Not Connected
E2	Vcc
E9	Vcc
E10	Gnd
F1	-SIZE
F2	GND (Reserved)
F9	D17
F10	D16
G1	-DS
G2	A4

Pin	Signal
G9	D20
G10	D18
H1	-AS
H2	A2
H3	A0
H8	Vcc
H9	GND (Reserved)
H10	D19
J1	A3
J2	Vcc (Reserved)

Pin	Signal
J3	-CS
J4	-DSACK0
J5	D31
J6	D28
J7	D25
J8	GND
J9	D23
J10	D21
K1	A1
K2	R/-W

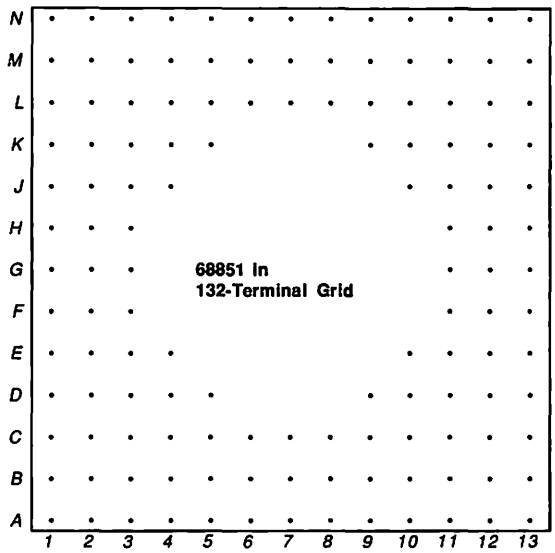
Pin	Signal
K3	GND
K4	-DSACK1
K5	D30
K6	D29
K7	D27
K8	D26
K9	D24
K10	D22

Notes: Minus sign (-) in front of signal name indicates negative going signal.

Source: MC68881/MC6882 Floating-Point Coprocessor User's Manual, Page 13-2

See Also: 4.43. 68000 Pinouts
4.44. 68020 Pinouts
4.46. 68851 Pinouts

4.46. 68851 Pinouts



Pin	Signal
A1	PA23
A2	PA29
A3	PA30
A4	-LAS
A5	-LBR0
A6	-LBGI
A7	-LBRI
A8	-CLI
A9	-PBG
A10	-PBGACK
A11	-DSACK0
A12	SIZ1
A13	-RESET

Pin	Signal
B1	PA19
B2	PA25
B3	PA26
B4	PA28
B5	-PAS
B6	-LBGACK
B7	Vcc
B8	-HALT
B9	-PBR
B10	SIZ0
B11	-DS
B12	-RMC
B13	D3

Pin	Signal
C1	PA18
C2	PA22
C3	CLK
C4	PA27
C5	PA31
C6	-LBG0
C7	GND
C8	-BERR
C9	-DSACK1
C10	R/-W
C11	DBDIS
C12	D0
C13	D4

Pin	Signal
D1	PA15
D2	PA20
D3	PA21
D4	PA24
D5	Vcc
D9	GND
D10	-ASYNC
D11	D1
D12	D2
D13	D7
E1	PA14
E2	PA16
E3	PA17

Pin	Signal
E4	GND
E10	Vcc
E11	D5
E12	D6
E13	D8
F1	PA11
F2	PA12
F3	PA13
F11	D9
F12	D10
F13	D11
G1	PA10
G2	Vcc

Pin	Signal
G3	GND
G11	GND
G12	Vcc
G13	D12
H1	PA9
H2	PA8
H3	A7
H11	D15
H12	D14
H13	D13
J1	A6
J2	A4
J3	A3

Pin	Signal
J4	Vcc
J10	GND
J11	D19
J12	D18
J13	D16
K1	A5
K2	A0
K3	FC3
K4	FC0
K5	GND
K9	Vcc
K10	D26
K11	D23

Pin	Signal
K12	D22
K13	D17

Pin	Signal
L1	A2
L2	FC2
L3	LA31
L4	LA28
L5	LA24
L6	LA20
L7	GND
L8	LA14
L9	LA10
L10	D30
L11	D27
L12	D24
L13	D20

Pin	Signal
M1	A1
M2	LA30
M3	LA29
M4	LA27
M5	LA23
M6	LA19
M7	Vcc
M8	LA15
M9	LA11
M10	D31
M11	D29
M12	D28
M13	D21

Pin	Signal
N1	FC1
N2	LA26
N3	LA25
N4	LA22
N5	LA21
N6	LA18
N7	LA17
N8	LA16
N9	LA13
N10	LA12
N11	LA9
N12	LA8
N13	D25

Notes: Minus sign (-) in front of signal name indicates negative going signal.

Source: MC68851 Paged Memory Management Unit User's Manual, page 13-1

See Also: 4.43. 68000 Pinouts
4.44. 68020 Pinouts
4.45. 68881 Pinouts

4.47. Single Inline Memory Module (SIMM Connector)

Macintosh Plus/SE

Pin #	*Signal Name*	*Signal Description*
1	+5V	Power
2	CAS1/	Column address strobe for SIMM1
	CAS2/	Column address strobe for SIMM2
	CAS3/	Column address strobe for SIMM3
	CAS4/	Column address strobe for SIMM4
3	RDQ0	RAM data bus bit 0 (SIMM1&3)
	RDQ8	RAM data bus bit 8 (SIMM 2&4)
4	RA0	RAM address bus bit 0
5	RA1	RAM address bus bit 1
6	RDQ1	RAM data bus bit 1 (SIMM 1&3)
	RDQ9	RAM data bus bit 9 (SIMM 2&4)
7	RA2	RAM address bus bit 2
8	RA3	RAM address bus bit 3
9	GND	Ground
10	RDQ2	RAM data bus bit 2 (SIMM 1&3)
	RDQ10	RAM data bus bit 10 (SIMM 2&4)
11	RA4	RAM address bus bit 4
12	RA5	RAM address bus bit 5
13	RDQ3	RAM data bus bit 3 (SIMM 1&3)
	RDQ11	RAM data bus bit 11 (SIMM 2&4)
14	RA6	RAM address bus bit 6
15	RA7	RAM address bus bit 7
16	RDQ4	RAM data bus bit 4 (SIMM 1&3)
	RDQ12	RAM data bus bit 12 (SIMM 2&4)
17	RA8	RAM address bus bit 8
18	RA9	RAM address bus bit 9
19	NOT USED	
20	RDQ5	RAM data bus bit 5 (SIMM 1&3)
	RDQ13	RAM data bus bit 13 (SIMM 2&4)
21	READ	RAM read
22	GND	Ground
23	RDQ6	RAM data bus bit 6 (SIMM 1&3)
	RDQ14	RAM data bus bit 14 (SIMM 2&4)
24	NOT USED	
25	RDQ7	RAM data bus bit 7 (SIMM 1&3)
	RDQ15	RAM data bus bit 15 (SIMM 2&4)
26	NOT USED	
27	RAS/	Row Address Strobe
28	NOT USED	
29	NOT USED	
30	+5V	Power

Macintosh II

Pin #	*Signal Name*	*Signal Description*
1	+5V	Power
2	/CASLL	Column address strobe for D0–D7
	/CASLM	Column address strobe for D8–D15
	/CASUM	Column address strobe for D16–D23
	/CASUU	Column address strobe for D24–D31
3	D0	RAM data bus bit 0
	D8	RAM data bus bit 8
	D16	RAM data bus bit 16
	D24	RAM data bus bit 24
4	RAAF0	RAM address bus, bank A bit 0
	RABF0	RAM address bus, bank B bit 0
5	RAAF1	RAM address bus, bank A bit 1
	RABF1	RAM address bus, bank B bit 1
6	D1	RAM data bus bit 1
	D9	RAM data bus bit 9
	D17	RAM data bus bit 17
	D25	RAM data bus bit 25
7	RAAF2	RAM address bus, bank A bit 2
	RABF2	RAM address bus, bank B bit 2
8	RAAF3	RAM address bus, bank A bit 3
	RABF3	RAM address bus, bank B bit 3
9	GND	Ground
10	D2	RAM data bus bit 2
	D10	RAM data bus bit 10
	D18	RAM data bus bit 18
	D26	RAM data bus bit 26
11	RAAF4	RAM address bus, bank A bit 4
	RABF4	RAM address bus, bank B bit 4
12	RAAF5	RAM address bus, bank A bit 5
	RABF5	RAM address bus, bank B bit 5
13	D3	RAM data bus bit 3
	D11	RAM data bus bit 11
	D19	RAM data bus bit 19
	D27	RAM data bus bit 27
14	RAAF6	RAM address bus, bank A bit 6
	RABF6	RAM address bus, bank B bit 6
15	RAAF7	RAM address bus, bank A bit 7
	RABF7	RAM address bus, bank B bit 7
16	D4	RAM data bus bit 4
	D12	RAM data bus bit 12
	D20	RAM data bus bit 20
	D28	RAM data bus bit 28
17	RAAF8	RAM address bus, bank A bit 8
	RABF8	RAM address bus, bank B bit 8
18	RAAF9	RAM address bus, bank A bit 9
	RABF9	RAM address bus, bank B bit 9
19	RAAF10	RAM address bus, bank A bit 10
	RABF10	RAM address bus, bank B bit 10
20	D5	RAM data bus bit 5
	D13	RAM data bus bit 13
	D21	RAM data bus bit 21
	D29	RAM data bus bit 29
21	RAMRWA	RAM read/write bank A
	RAMRWB	RAM read/write bank B
22	GND	Ground
23	D6	RAM data bus bit 6
	D14	RAM data bus bit 14
	D22	RAM data bus bit 22
	D30	RAM data bus bit 30
24	RAAF11	RAM address bus, bank A bit 11
	RABF11	RAM address bus, bank B bit 11
25	D7	RAM data bus bit 7
	D15	RAM data bus bit 15
	D23	RAM data bus bit 23
	D31	RAM data bus bit 31
26	NOT USED	
27	/RASA	Row Address Strobe bank A
	/RASB	Row Address Strobe bank B
28	PU	Pull up
29	NOT USED	
30	+5V	Power

Source: Apple Macintosh Family Hardware Reference, pages 3-7, 10-6 to 10-7, and 18-7 to 18-8

4.48. Apple DeskTop Bus Pinouts

Pin #	*Signal Name*	*Signal Description*
1	ADB	Bidirectional bit-at-a-time data bus
2	NOT USED	
3	+5V	Power
4	GND	Logic ground

Notes: Connector is a 4-pin circular connector with key.

Source: Apple Macintosh Family Hardware Reference, pages 11-4 to 11-5

4.49. NuBus Card Layout

Thickness	0.062 inches, plus or minus 0.0075 inches
Height	4.0 inches
Length	Between 7.0 and 12.875 inches
Component height	No higher than 0.60 inches, measured from the card surface
Connector	Euro-DIN 603-2-IEC-C096-M, 96-contacts, 2.54mm spacing

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 7-2 to 7-3

See Also: 4.50. NuBus Bus Signals (Connector)

4.50. NuBus Bus Signals (Connector)

Pin #	Signal Name	Signal Description
A1	-12V	Power
A2	GND	Ground
A3	/SPV	System parity valid
A4	/SP	System parity
A5	/TM1	Transfer mode 1
A6	/AD1	Address/data
A7	/AD3	Address/data
A8	/AD5	Address/data
A9	/AD7	Address/data
A10	/AD9	Address/data
A11	/AD11	Address/data
A12	/AD13	Address/data
A13	/AD15	Address/data
A14	/AD17	Address/data
A15	/AD19	Address/data
A16	/AD21	Address/data
A17	/AD23	Address/data
A18	/AD25	Address/data
A19	/AD27	Address/data
A20	/AD29	Address/data
A21	/AD31	Address/data
A22	GND	Ground
A23	GND	Ground
A24	/ARB1	Arbitration 1
A25	/ARB3	Arbitration 3
A26	/ID1	Slot identification 1
A27	/ID3	Slot identification 3
A28	/ACK	Acknowledge
A29	+5V	Power
A30	/RQST	Request
A31	/NMRQ	Non-master request
A32	+12V	Power

Pin #	Signal Name	Signal Description
B1	-12V	Power
B2	GND	Ground
B3	GND	Ground
B4	+5V	Power
B5	+5V	Power
B6	+5V	Power
B7	+5V	Power
B8	NC	No connection *
B9	NC	No connection *
B10	NC	No connection *
B11	NC	No connection *
B12	GND	Ground
B13	GND	Ground
B14	GND	Ground
B15	GND	Ground
B16	GND	Ground
B17	GND	Ground
B18	GND	Ground
B19	GND	Ground
B20	GND	Ground
B21	GND	Ground
B22	GND	Ground
B23	GND	Ground
B24	NC	No connection *
B25	NC	No connection *
B26	NC	No connection *
B27	NC	No connection *
B28	+5V	Power
B29	+5V	Power
B30	GND	Ground
B31	GND	Ground
B32	+12V	Power

Pin #	Signal Name	Signal Description
C1	/RESET	Reset
C2	GND	Ground
C3	+5V	Power
C4	+5V	Power
C5	/TM0	Transfer mode 0
C6	/AD0	Address/data
C7	/AD2	Address/data
C8	/AD4	Address/data
C9	/AD6	Address/data
C10	/AD8	Address/data
C11	/AD10	Address/data
C12	/AD12	Address/data
C13	/AD14	Address/data
C14	/AD16	Address/data
C15	/AD18	Address/data
C16	/AD20	Address/data
C17	/AD22	Address/data
C18	/AD24	Address/data
C19	/AD26	Address/data
C20	/AD28	Address/data
C21	/AD30	Address/data
C22	GND	Ground
C23	/PFW	Power fail warning
C24	/ARB0	Arbitration 0
C25	/ARB2	Arbitration 2
C26	/ID0	Slot identification 0
C27	/ID2	Slot identification 2
C28	/START	Start
C29	+5V	Power
C30	+5V	Power
C31	GND	Ground
C32	/CLK	Clock

Notes:
- All signals refer to Apple's definition of NuBus only.
- Connector is a 603-2-IEC-C096-M.
- * These pins are connected, but not supplied with -5.2V. This voltage may be supplied by a card.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 2-5, 6-5

See Also: 4.49. NuBus Card Layout

4.51. SE Expansion Card Layout

Thickness	0.063 inches
Height	4.335 inches (4.741 inches counting tab)
Length	8.396 inches
Component height	Maximum free space 0.181 inches
Connector	Euro-DIN, 96-contacts, 2.54mm spacing

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 14-3 to 14-11

See Also: 4.52. SE Bus Expansion Signals (Connector)

4.52. SE Bus Expansion Signals (Connector)

Pin #	Signal Name	Signal Description
A1	FC2	Function code 2
A2	FC1	Function code 1
A3	FC0	Function code 0
A4	A1	Address
A5	A2	Address
A6	A3	Address
A7	A4	Address
A8	A5	Address
A9	A6	Address
A10	A7	Address
A11	A8	Address
A12	A9	Address
A13	A10	Address
A14	A11	Address
A15	A12	Address
A16	A13	Address
A17	A14	Address
A18	A15	Address
A19	A16	Address
A20	A17	Address
A21	A18	Address
A22	A19	Address
A23	A20	Address
A24	A21	Address
A25	A22	Address
A26	A23	Address
A27	E	Enable clock
A28	C8M	7.8336 Mhz clock
A29	C16M	15.6672 Mhz clock
A30	GND	Logic ground
A31	+12V	Power
A32	+12V	Power

Pin #	Signal Name	Signal Description
B1	GND	Ground
B2	GND	Ground
B3	GND	Ground
B4	GND	Ground
B5	GND	Ground
B6	GND	Ground
B7	GND	Ground
B8	GND	Ground
B9	GND	Ground
B10	RESERVED	
B11	RESERVED	
B12	/HALT	68000 halt
B13	+5V	Power
B14	+5V	Power
B15	+5V	Power
B16	+5V	Power
B17	+5V	Power
B18	/IPL0	Int level 0 (VIA,SCSI)
B19	/IPL1	Int level 1 (SCC)
B20	/IPL2	Int level 2 (NMI)
B21	/BERR	Bus error
B22	NC	No connection
B23	RESERVED	
B24	RESERVED	
B25	RESERVED	
B26	RESERVED	
B27	RESERVED	
B28	/EXT.DTK	External /DTACK
B29	GND	Ground
B30	+12V	Power
B31	+12V	Power
B32	-5V	Power

Pin #	Signal Name	Signal Description
C1	/VPA	Valid periph addr
C2	/VMA	Valid mem addr
C3	/BR	Bus request
C4	/BGACK	Bus grant ackn
C5	/BG	Bus grant
C6	/DTACK	Data trans ackn
C7	R/W	Read/write
C8	/LDS	Lower data strobe
C9	/UDS	Upper data strobe
C10	/AS	Address strobe
C11	/PMCYC	Proc mem cycle
C12	/RESET	System reset
C13	+5V	Power
C14	D0	Data
C15	D1	Data
C16	D2	Data
C17	D3	Data
C18	D4	Data
C19	D5	Data
C20	D6	Data
C21	D7	Data
C22	D8	Data
C23	D9	Data
C24	D10	Data
C25	D11	Data
C26	D12	Data
C27	D13	Data
C28	D14	Data
C29	D15	Data
C30	GND	Ground
C31	NC	No connection
C32	-12V	Power

Notes:
- For loading/driving capability, see source.
- Connector is Euro-DIN 96-pin socket connector.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, pages 13-2 to 13-9

See Also: 4.51. SE Expansion Card Layout

4.53. LaserWriter NTX Card Layout

Thickness	No particular guidelines
Height	5.25 inches maximum
Length	4.5 inches maximum
Component height	No particular guidelines
Connector	Euro-DIN, 96-contacts, 2.54mm spacing

Source: LaserWriter Reference, pages 128 to 129

See Also: 4.54. LaserWriter NTX Bus Signals

4.54. LaserWriter NTX Bus Signals (Connector)

Pin #	Signal Name	Signal Description
A1	NC	No connection
A2	A2	Address 2
A3	A3	Address 3
A4	A4	Address 4
A5	A5	Address 5
A6	A6	Address 6
A7	A7	Address 7
A8	A8	Address 8
A9	A9	Address 9
A10	A10	Address 10
A11	A11	Address 11
A12	A12	Address 12
A13	A13	Address 13
A14	A14	Address 14
A15	A15	Address 15
A16	A16	Address 16
A17	A17	Address 17
A18	A18	Address 18
A19	A19	Address 19
A20	A20	Address 20
A21	A21	Address 21
A22	A22	Address 22
A23	A23	Address 23
A24	A24	Address 24
A25	A25	Address 25
A26	A26	Address 26
A27	A27	Address 27
A28	A28	Address 28
A29	A29	Address 29
A30	A30	Address 30
A31	A31	Address 31
A32	A32	Address 32

Pin #	Signal Name	Signal Description
B1	+5V	Power
B2	+5V	Power
B3	+5V	Power
B4	+5V	Power
B5	+5V	Power
B6	+5V	Power
B7	+5V	Power
B8	+5V	Power
B9	NC	No connection
B10	NC	No connection
B11	NC	No connection
B12	NC	No connection
B13	NC	No connection
B14	NC	No connection
B15	NC	No connection
B16	NC	No connection
B17	dbrom0n	Data buffer select 0
B18	dbrom1n	Data buffer select 1
B19	dbrom2n	Data buffer select 2
B20	dbrom3n	Data buffer select 3
B21	NC	No connection
B22	NC	No connection
B23	NC	No connection
B24	NC	No connection
B25	GND	Ground
B26	GND	Ground
B27	GND	Ground
B28	GND	Ground
B29	GND	Ground
B30	GND	Ground
B31	GND	Ground
B32	GND	Ground

Pin #	Signal Name	Signal Description
C1	D0	Data 0
C2	D1	Data 1
C3	D2	Data 2
C4	D3	Data 3
C5	D4	Data 4
C6	D5	Data 5
C7	D6	Data 6
C8	D7	Data 7
C9	D8	Data 8
C10	D9	Data 9
C11	D10	Data 10
C12	D11	Data 11
C13	D12	Data 12
C14	D13	Data 13
C15	D14	Data 14
C16	D15	Data 15
C17	D16	Data 16
C18	D17	Data 17
C19	D18	Data 18
C20	D19	Data 19
C21	D20	Data 20
C22	D21	Data 21
C23	D22	Data 22
C24	D23	Data 23
C25	D24	Data 24
C26	D25	Data 25
C27	D26	Data 26
C28	D27	Data 27
C29	D28	Data 28
C30	D29	Data 29
C31	D30	Data 30
C32	D31	Data 31

Notes: Connector is a Euro-DIN, 96-pin, 2.54mm spacing.

Source: LaserWriter Reference, page 130

See Also: 4.53. LaserWriter NTX Card Layout

4.55 Macintosh 128/512 DB-9 Serial Port Connectors

Pin #	*Signal Name*	*Signal Description*	*RS-232 Equivalent Signal**
1	SG	Signal ground	Shield
2	+5V	Power	-
3	SG	Signal ground	Signal ground
4	TxD+	Transmitted data +	Data carrier detect (DCD)
5	TxD-	Transmitted data -	Received data (RxD)
6	+12V	Power	-
7	HSKo	Handshake/ext clock	Data terminal ready (DTR)
8	RxD+	Received data +	No equivalent
9	RxD-	Received data -	Transmitted data (TxD)

Notes:
- All signals refer to RS-422 assignments.
- Do not draw more than 100ma at +12 volts, or 200ma at +5 volts.
- * Assumes RS-232 device is configured as a DCE device.

Source:
Apple LaserWriter Reference, page 141
Inside Macintosh, page III-24

See Also:
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors
4.57. ImageWriter DB-9 and DB-25 Serial Port Connector
4.58. LaserWriter DB-9 Serial Port Connector
4.59. LaserWriter Mini-8 Serial Port Connector
4.60. LaserWriter DB-25 Serial Port Connector
4.61. Apple Modem Serial Port Connectors

4.56 Macintosh Plus/SE/II Mini-8 Serial Port Connectors

Pin #	*Signal Name*	*Signal Description*	*RS-232 Equivalent Signal**
1	HSKo	Output handshake	Data terminal ready (DTR)
2	HSKi	Input handshake/ext clock	Clear to send (CTS)
3	TxD-	Transmitted data -	Received data (RxD)
4	GND	Ground	Ground (GND)
5	RxD-	Received data -	Transmitted data (TxD)
6	TxD+	Transmitted data +	Data carrier detect (DCD)
7	-	NOT USED	
8	RxD+	Received data +	No equivalent

Notes:
- All signals refer to RS-422 assignments.
- Connector is a Mini-8 socket.
- *Assumes RS-232 device is configured as a DCE device.

Source:
Apple LaserWriter Reference, page 141
Inside Macintosh, pages IV-248 to IV-249

See Also:
4.55. Macintosh 128/512 DB-9 Serial Port Connectors
4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors
4.58. LaserWriter DB-9 Serial Port Connector
4.59. LaserWriter Mini-8 Serial Port Connector
4.60. LaserWriter DB-25 Serial Port Connector
4.61. Apple Modem Serial Port Connectors

4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors

DB-25 on ImageWriter I

Pin #	*Signal Name*	*Signal Description*
1	Shield	Protective/chassis ground
2	TxD	Transmitted data (out)
3	RxD	Received data (in)
4	RTS	Request to send (out)
5	-	NOT USED
6	-	NOT USED
7	GND	Signal ground
8	-	NOT USED
9	-	NOT USED
10	-	NOT USED
11	-	NOT USED
12	-	NOT USED
13	-	NOT USED
14	FAULT	Fault (printer not selected) (out)
15	-	NOT USED
16	-	NOT USED
17	-	NOT USED
18	-	NOT USED
19	-	NOT USED
20	DTR	Data terminal ready (out)
21	-	NOT USED
22	-	NOT USED
23	-	NOT USED
24	-	NOT USED
25	-	NOT USED

Mini-8 on ImageWriter II

Pin #	*Signal Name*	*Signal Description*
1	DTR	Data terminal ready (out)
2	DSR	Data set ready (in)
3	TxD-	Transmitted data (out)
4	GND	Signal ground
5	RxD-	Received data (in)
6	TxD+	Balanced transmit (out)
7	NC	NO CONNECTION
8	RxD+	Balanced receive (in)
Shield	GND	Chassis ground

Notes: *ImageWriter I:*
- All signals refer to RS-232C assignments.
- Connector is a DB-25 socket.

ImageWriter II:
- All signals refer to RS422 assignments.
- Connector is a Mini-8 socket.

Source:
Apple ImageWriter User's Manual, pages 103 to 104
ImageWriter II Technical Reference Manual, pages 188 to 189

See Also:
4.55. Macintosh 128/512 DB-9 Serial Port Connectors
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors
4.58. LaserWriter DB-9 Serial Port Connector
4.59. LaserWriter Mini-8 Serial Port Connector
4.60. LaserWriter DB-25 Serial Port Connector
4.61. Apple Modem Serial Port Connectors

4.58 LaserWriter DB-9 Serial Port Connector

Pin #	*Signal Name*	*Signal Description*	*RS-232 Equivalent Signal**
1	SG	Signal ground	Signal ground
2	-	NOT USED	-
3	SG	Signal ground	Signal ground
4	TxD+	Transmitted data +	Data carrier detect (DCD)
5	TxD-	Transmitted data -	Received data (RxD)
6	-	NOT USED	-
7	-	NOT USED	-
8	RxD+	Received data +	No equivalent
9	RxD-	Received data -	Transmitted data (TxD)

Notes:

- All signals refer to RS-422 assignments.
- * Assumes RS-232 device is configured as a DCE device.

Source:

Apple LaserWriter Reference, pages 139, 141

See Also:

4.55. Macintosh 128/512 DB-9 Serial Port Connectors
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors
4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors
4.59. LaserWriter Mini-8 Serial Port Connector
4.60. LaserWriter DB-25 Serial Port Connector
4.61. Apple Modem Serial Port Connectors

4.59 LaserWriter Mini-8 Serial Port Connector

Pin #	*Signal Name*	*Signal Description*	*SCC Connection*	*RS-232 Equivalent Signal**
1	HSKo	Output handshake	SCC data terminal ready	Data terminal ready (DTR)
2	HSKi	Input handshake/ext clock	SCC transmit/receive clock	Clear to send (CTS)
3	TxD-	Transmitted data -	SCC transmit data	Received data (RxD)
4	GND	Ground	SCC logic and chassis ground	Ground
5	RxD-	Received data -	SCC receive data	Transmitted data (TxD)
6	TxD+	Transmitted data +	SCC transmit data	Data carrier detect (DCD)
7	-	NOT USED	-	-
8	RxD+	Received data +	SCC receive data	No equivalent

Notes:
- All signals refer to RS-422 assignments.
- Connector is a Mini-8 socket.
- *Assumes RS-232 device is configured as a DCE device.

Source: Apple LaserWriter Reference, page 139

See Also:
4.55. Macintosh 128/512 DB-9 Serial Port Connectors
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors
4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors
4.58. LaserWriter DB-9 Serial Port Connector
4.60. LaserWriter DB-25 Serial Port Connector
4.61. Apple Modem Serial Port Connectors

4.60. LaserWriter DB-25 Serial Port Connector

Pin #	*Signal Name*	*Signal Description*
1	Shield	Protective/chassis ground
2	TxD	Transmitted data
3	RxD	Received data
4	RTS	Request to send
5	CTS	Clear to send
6	DSR	Data set ready
7	GND	Signal ground
8	DCD	Data carrier detect
9	-	NOT USED
10	-	NOT USED
11	-	NOT USED
12	-	NOT USED
13	-	NOT USED
14	-	NOT USED
15	-	NOT USED
16	-	NOT USED
17	-	NOT USED
18	-	NOT USED
19	-	NOT USED
20	DTR	Data terminal ready *
21	-	NOT USED
22	RI	Ring indicator
23	-	NOT USED
24	-	NOT USED
25	-	NOT USED

(continued)

Table 4.60. Continued

Notes:
- All signals refer to RS-232C assignments.
- Connector is a DB-25 socket.
- * Power on terminal signal for LaserWriter II NTX.

Source: Apple LaserWriter Reference, page 140

See Also:
4.55. Macintosh 128/512 DB-9 Serial Port Connectors
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors
4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors
4.58. LaserWriter DB-9 Serial Port Connector
4.59. LaserWriter Mini-8 Serial Port Connector
4.61. Apple Modem Serial Port Connectors

4.61. Apple Modem Serial Port Connectors

Apple Modem 300/1200 (DB-9)

Pin #	*Signal Name*	*Signal Description*
1	NC	No connection
2	DSR	Data Set Ready (out)
3	GND	Signal ground
4	RxD	Receive data (out)
5	NC	No connection
6	DTR	Data terminal ready (in)
7	DCD	Data carrier detect (out)
8	GND	Chassis ground
9	TxD	Transmit data (in)

Apple Personal Modem (Mini-8)

1	DSR	Data Set Ready (out)
2	DTR	Data terminal ready (in)
3	RxD	Receive data (out)
4	GND	Signal ground
5	TxD	Transmit data (in)
6	GND	Signal ground
7	DCD	Data carrier detect (out)
8	NC	No connection

Notes:
- Both modems are labeled DCE but are really DTE (see I/O in parens).
- Connectors are DB-9 and Mini-8, respectively.

Source: BMUG Fall/Winter 1987 Newsletter, pages 110 to 126

See Also:
4.55. Macintosh 128/512 DB-9 Serial Port Connectors
4.56. Macintosh Plus/SE/II Mini-8 Serial Port Connectors
4.57. ImageWriter DB-9 and DB-25 Serial Port Connectors
4.58. LaserWriter DB-9 Serial Port Connector
4.59. LaserWriter Mini-8 Serial Port Connector
4.60. LaserWriter DB-25 Serial Port Connector

4.62. Macintosh 128/512/Plus/SE External Drive Connector

Pin #	*Signal Name*	*Signal Description*
1	GND	Signal ground
2	GND	Signal ground
3	GND	Signal ground
4	GND	Signal ground
5	-12V	Power (10ma max load)
6	+5V	Power (200ma maxload)
7	+12V	Power (100ma max load)
8	+12V	Power (100ma max load)
9	-	NOT USED
10	MSC	Motor signal control
11	CA0	Disk state control line 0
12	CA1	Disk state control line 1
13	CA2	Disk state control line 2
14	LSTRB	Line strobe
15	WR	Write request
16	SEL	Select
17	DE	External drive enable
18	RD	Read data
19	WD	Write data

Notes:
- Maximum loads on power are for all connectors on system.
- Connector is a DB-19 socket.

Source: Inside Macintosh, page III-33

See Also: 4.22. Disk State Control Lines and Registers

4.63. Macintosh Plus/SE/II SCSI Connector

Pin #	*Signal Name*	*Signal Description*
1	-REQ	Request
2	-MSG	Message
3	-I/O	I/O direction
4	-RST	Reset
5	-ACK	Acknowledge
6	-BSY	Busy
7	GND	Ground
8	-DB0	Data bit 0
9	GND	Ground
10	-DB3	Data bit 3
11	-DB5	Data bit 5
12	-DB6	Data bit 6
13	-DB7	Data bit 7
14	GND	Ground
15	-C/D	Control or data
16	GND	Ground
17	-ATN	Attention
18	GND	Ground
19	-SEL	Select
20	-DBP	Data bit (parity)
21	-DB1	Data bit 1
22	-DB2	Data bit 2
23	-DB4	Data bit 4
24	GND	Ground
25	-	NOT USED

Notes: Connector is a DB-25 socket.

Source: Inside Macintosh, page IV-252
Macintosh Family Hardware Reference, page 5-21

See Also: 4.21. SCSI Address Assignments
3.234. to 3.239. SCSI Manager (Operating System Toolkit) tables

4.64. Sound Pinouts

Macintosh Plus/SE

Pin #	*Signal Name*	*Signal Description*
Plug	SND	8V peak-to-peak, low impedance signal
Shell	GND	Ground

Macintosh II

Pin #	*Signal Name*	*Signal Description*
Tip	CHNL 2	1.5V peak-to-peak, 47 ohm source impedance, channel 2
Plug	CHNL 1	1.5V peak-to-peak, 47 ohm source impedance, channel 1
Shell	GND	Ground

Notes:
- Macintosh Plus/SE: Connector is a monophonic microphone jack.
- Macintosh II: Connector is a stereo microphone jack.

Source: Apple Macintosh Family Hardware Reference, pages 13-5, 21-5

4.65. Macintosh 128/512/Plus Keyboard Pinouts

Pin #	*Signal Name*
1	Ground
2	Clock
3	Data
4	+5V

Notes:
- Connector is RJ-11 telephone style jack/plug.
- Do not draw more than 200ma at +5V from all connectors.

Source: Inside Macintosh, pages III-29 to III-30

See Also: 4.48. Apple DeskTop Bus Pinouts

4.66. Macintosh 128/512/Plus Mouse Pinouts

Pin #	*Signal Name*	*Comments*
1	Ground	
2	+5V	
3	Ground	
4	Mouse X2	VIA quadrature signal
5	Mouse X1	SCC interrupt signal
6	NO CONNECTION	
7	Mouse Switch	
8	Mouse Y2	VIA quadrature signal
9	Mouse Y1	SCC interrupt signal

Notes:
- Connector is DB-9 socket.
- Do not draw more than 200ma at +5V from all connectors.

Source: Inside Macintosh, page III-28

See Also: 4.48. Apple DeskTop Bus Pinouts

4.67. Macintosh II Video Card Pinouts

Pin #	*Signal Name*	*Signal Description*
1	GND	Ground
2	RED	Red color signal
3	/CSYNCH	Synchronization
4	GND	Ground
5	GREEN	Green color signal
6	GND	Ground
7	NC	No connection
8	NC	No connection
9	BLUE	Blue color signal
10	NC	No connection
11	GND	Ground
12	NC	No connection
13	GND	Ground
14	GND	Ground
15	NC	No connection

Notes:
- Signals are for analog monitor.
- Connector is DB-15 socket.

Source: Designing Cards and Drivers for the Macintosh II and Macintosh SE, page 11-18

4.68. ImageWriter Family History

	ImageWriter I	*ImageWriter II*	*ImageWriter LQ*
Width	15.7" (398mm)	17.0" (431.8mm)	23.0" (584.2mm)
Depth	11.3" (285mm)	12.0" (304.8mm)	15.0" (381mm)
Height	5.3" (125mm)	5.0" (127mm)	5.0" (127mm)
Weight	18.75 lbs (8.5kg)	25 lbs (11.36kg)	38 lbs (17.1kg)
Print Speed	120 cps (draft)	250 cps (draft), 45 cps (NLQ)	250 cps (draft), 90 cps (NLQ)
Char Format	8x7 (standard), 8x16 (custom)	7x12 (draft), 14x16 (NLQ)	7x12 (draft), 24x16 (NLQ)
Graphics Res.	72 dpi	72 dpi	72 dpi, 144 dpi, 216 dpi
Ports	RS-232C	RS-422, 423, 232*	RS-422, 423, 232*
Char Pitches	4.5,5,6,6.7,7.5,8.5,9,10,12,13.4,15,17 cpi + prop.	9, 10, 12, 13.4, 15, 17 cpi + prop.	9, 10, 12, 13.4, 15, 17 cpi + prop.
Paper Types	Single sheets, roll paper, pin-feed 4" to 9.5"	Single sheets, pin-feed 3.5" to 9.5"	Single sheets, pin-feed 3.5" to 15", sheet feeder
Ribbon	Black	Black, four-color	Black, four-color
Introduced	Feb-84	Sep-85	Jan-88

Notes: * Although the printer was designed to support RS-422, a subset of RS-232C may be used to drive the printer.

Source: Apple ImageWriter II Reference, pages 184 to 185
Apple ImageWriter User's Manual, pages 101 to 102
Apple ImageWriter LQ Reference, pages 223 to 225, 227 to 228

See Also: 4.77. LaserWriter Family History

4.69. ImageWriter Default Settings

	ImageWriter I	*ImageWriter II*	*ImageWriter LQ*
Color Option	NA	ON if color ribbon installed	ON if color ribbon installed
Communications Rate	Per DIP switches 2-1,2-2	Per DIP switches 2-1, 2-2	Per DIP switches 2-1, 2-2
Communications Protocol	Per DIP switch 2-3	Per DIP switch 2-3	Per DIP switch 2-3
Option Card Enabled	NA	Per DIP switch 2-4	Per DIP switch 2-4
Font	NA	Draft	Draft
Character Pitch	Elite	Per DIP switches 1-6, 1-7	Per DIP switches 1-6, 1-7
Proportional Character Spacing	1 dot width	0	0
Slashed/Unslashed Zeroes	NA	Unslashed	Unslashed
Select MouseText Characters	NA	NO	NA
Underline Text	OFF	OFF	OFF
Boldface Text	OFF	OFF	OFF
Double-width Text	OFF	OFF	OFF
Half-height Text	NS	OFF	OFF
Subscript Text	NS	OFF	OFF
Superscript Text	NS	OFF	OFF
Left Margin	0	0	0
Top of Form	Position of print head on powerup	Position of print head on powerup	Position of print head on powerup
Page Length	Per DIP switch 1-4	Per DIP switch 1-4	Per DIP switch 1-4
Paper-Out Sensor	ON	ON	ON
Cut Sheet Feeder Bins	NA	NA	Per DIP switches 2-5, 2-6, 2-7
Auto Paper Load Position	NA	NA	Per DIP switch 2-8
Line Spacing	6 lines per inch	6 lines per inch	6 lines per inch
Line Feed Direction	Forward	Forward	Forward
Carriage Return Insertion	NS	ON	ON
Line Feed After Carriage Return	Per DIP switch 1-8	Per DIP switch 1-8	Per DIP switch 1-8
Line Feed At Line Buffer Full	NS	OFF	OFF
Software Select Response	NA	Disabled	Disabled
Commands That Cause Printing	NA	CR, LF, FF	CR, LF, FF
Perforation Skip	NS	Per DIP switch 1-5	Per DIP switch 1-5
8th Bit Support	Per DIP switch 1-5	IGNORED	IGNORED
Print Head Motion	Bidirectional	Bidirectional	Bidirectional
Horizontal Tabs	Cleared	Cleared	Cleared
Vertical Tabs	B tab every 6 lines	NA	NA
Graphics Printing	OFF	OFF	OFF
Ribbon Color	NA	Black	Black
Custom Characters	OFF	Cleared (powerup), Saved (soft reset)	NA

Notes: NA = not applicable, NS = not specified in manual.

Source: Apple ImageWriter User's Manual, pages 51, 81 to 82
ImageWriter II Technical Reference Manual, pages 134 to 135
ImageWriter LQ Technical Reference Manual, pages 27 to 28, 164 to 175

See Also: 4.70. ImageWriter II Software Switches
4.75. ImageWriter DIP Switch Settings

4.70. ImageWriter II Software Switches

Bit Number

7	6	5	4	3	2	1	0	*Switch A**	*Switch B***
0	0	0	0	0	0	0	X	Language Set 1**	Use Slashed Zeros
0	0	0	0	0	0	X	0	Language Set 2**	NA
0	0	0	0	0	X	0	0	Language Set 3**	Set Perforation Skip
0	0	0	0	X	0	0	0	NA	NA
0	0	0	X	0	0	0	0	Software Select Response	NA
0	0	X	0	0	0	0	0	LF When Line Is Full	Use 8th Data Bit
0	X	0	0	0	0	0	0	Commands That Cause Print	NA
X	0	0	0	0	0	0	0	LF after Every CR	NA

Notes:
- NA = not applicable (cannot set from software).
- To use: Set the bits that correspond to the switch(es) you want to set when using the ESC Z a b or ESC D a b software commands. The Z command sets switches to open, D to closed. (See 4.75. ImageWriter DIP Switch Settings for more information on how these switches are set.)
- * Software Switch A corresponds to Hardware Switch 1, B to Switch 2.
- ** These three switches are set together to select character set.

Source: ImageWriter II Technical Reference Manual, pages 31 to 32

See Also: 4.75. ImageWriter DIP Switch Settings

4.71. ImageWriter Printer Control Functions

Character Sets

Command	*Form*	*Hexadecimal*	*Decimal*
Select Correspondence Font §	ESC a 0	$1B $61 $30	27 97 48
Select Draft Font ‡	ESC a 1	$1B $61 $31	27 97 49
Select NLQ Font	ESC a 2	$1B $61 $32	27 97 50
Select LQ Font ¥	ESC a 3	$1B $61 $33	27 97 51
Send Name of NLQ Font ¥	ESC = 0	$1B $3D $30	27 61 48
Send Name of Draft Font ¥	ESC = 1	$1B $3D $31	27 61 49
Send Name of NLQ Font ¥	ESC = 2	$1B $3D $32	27 61 50
Send Name of LQ Font ¥	ESC = 3	$1B $3D $33	27 61 51
Send Name of Optional Font ¥	ESC = 9	$1B $3D $39	27 61 57
Select Optional Font ¥	ESC k	$1B $6B	27 107
Select Correspondence Font ****	ESC m	$1B $6D	27 109
Select NLQ Font ****	ESC M	$1B $4D	27 77
Map MouseText Chars to low ASCII ****	ESC &	$1B $26	27 38
Select Standard Chars to low ASCII **** ‡	ESC $	$1B $24	27 36

Software Switch Setting

Set Switches Open * †	ESC Z (a) (b)	$1B $5A (a) (b)	27 90 (a) (b)
Set Switches Closed * †	ESC D (a) (b)	$1B $44 (a) (b)	27 68 (a) (b)

Custom Characters

Set Custom Width to 8 **** ‡	ESC -	$1B $2D	27 45
Set Custom Width to 16 ****	ESC +	$1B $2B	27 43
Load Custom Characters ****	ESC I (data) Ctrl-D	$1B $49 (data) $04	27 73 (data) 4
Switch to Custom Character Font ****	ESC '	$1B $27	27 39
Switch to Custom Char Font (Hi ASCII) ****	ESC *	$1B $2A	27 42
Switch to Normal Font ****	ESC $	$1B $24	27 36

(continued)

Table 4.71. Continued

Graphics

Command	*Form*	*Hexadecimal*	*Decimal*
Print # Bytes Graphics on Line	ESC G (#) (data)	$1B $47 (#) (data)	27 71 (#) (data)
Print #*8 Bytes Graphics on Line	ESC g (#) (data)	$1B $67 (#) (data)	27 103 (#) (data)
High Res Print Head Placement ¥	ESC h (#)	$1B $68 (#)	27 104 (#)
High Res Graphics ¥	ESC C (#)	$1B $43 (#)	27 67 (#)
Vertical Dot Shift ¥	ESC t (#)	$1B $74 (#) [also $1B $4C]	27 116 (#) [also 27 76]
High Res Repeating Dot Columns ¥	ESC U (#) (data)	$1B $55 (#) (data)	27 85 (#) (data)
Print # Repetitions of Graphics	ESC V (#) (data)	$1B $56 (#) (data)	27 86 (#) (data)
Place Print Head in Dot Column	ESC F (column)	$1B $46 (column)	27 70 (column)

Character Enhancements

Start Double-width Printing	Ctrl-N	$0E	14
Stop Double-width Printing	Ctrl-O	$0F	15
Set Ribbon **	ESC K (color)	$1B $4B (color)	27 75 (color)
Repeat Character # Times	ESC R (#) (char)	$1B $52 (#) (char)	27 82 (#) (char)
Set 9 cpi - Extended Chars	ESC n	$1B $6E	27 110
Set 10 cpi - Pica Chars	ESC N	$1B $4E	27 78
Set 12 cpi - Elite Chars	ESC E	$1B $45	27 69
Set 13.4 cpi - Semicondensed Chars	ESC e	$1B $65	27 101
Set 15 cpi - Condensed Chars	ESC g	$1B $71	27 113
Set 17 cpi - Ultracondensed Chars	ESC Q	$1B $51	27 81
Set 144 dpi - Pica Prop. Chars	ESC p	$1B $70	27 112
Set 160 dpi - Elite Prop. Chars	ESC P	$1B $50	27 80
Set Dot Spacing to # for All Chars	ESC s (#)	$1B $73 (#: 0–9)	27 115 (#: 0–9)
Insert # Dot Spaces Between Chars	ESC (#)	$1B (#: 1–6)	27 (#: 1–6)
Start Underline	ESC X	$1B $58	27 88
Stop Underline ‡	ESC Y	$1B $59	27 89
Start Boldface	ESC !	$1B $21	27 33
Stop Boldface ‡	ESC "	$1B $22	27 34
Start Half-Height	ESC w	$1B $77	27 119
Stop Half-Height ‡	ESC W	$1B $57	27 87
Start Superscript	ESC x	$1B $78	27 120
Start Subscript	ESC y	$1B $79	27 121
Stop Sub or Superscript ‡	ESC z	$1B $7A	27 122

Page and Paper Settings

Set Left Margin to Column ***	ESC L (#)	$1B $4C (#)	27 76 (#)
Set Page Length to #/144 inches	ESC H (#: 1 to 9999)	$1B $48 (#)	27 72 (#)
Line Feed	Ctrl-J	$0A	10
Multiple Line Feed****	Ctrl-_ (#: 1 to 9 : ; < = >)	$1F (#)	31 (#)
6 Lines per Inch	ESC A	$1B $41	27 65
8 Lines per inch	ESC B	$1B $42	27 66
Distance Between Lines (#/144")	ESC T (#: 1 to 99)	$1B $54 (#)	27 84 (#)
Forward Paper Feed ‡	ESC f	$1B $66	27 102
Reverse Paper Feed	ESC r	$1B $72	27 114
Use First CSF Bin ¥	ESC @ 0	$1B $40 $30	27 64 48
Use Second CSF Bin ¥	ESC @ 1	$1B $40 $31	27 64 49
Use Third CSF Bin ¥	ESC @ 2	$1B $40 $32	27 64 50
Form Feed	Ctrl-L	$0C	12
Set Top of Form to Current Position	ESC v	$1B $76	27 118
Paper-out Sensor OFF	ESC O	$1B $4F	27 79
Paper-out Sensor ON ‡	ESC o	$1B $6F	27 111

Print Head Motion

Carriage Return	Ctrl-M	$0D	13
Unidirectional printing	ESC >	$1B $3E	27 62
Bidirectional printing ‡	ESC <	$1B $3C	27 60
Backspace	Ctrl-H	$08	8
Set Horizontal Tabs ***	ESC ((#),....(#).	$1B $28 (#) $2C...$2C (#) $2E	27 40 (#) 44...44(#) 46
Add Tab Stop ***	ESC u (#)	$1B $75 (#)	27 117 (#)
Move Print Head to Next Tab Stop	Ctrl-I	$09	9
Clear Horizontal Tabs	ESC) (#),....(#).	$1B $29 (#) $2C...$2C (#) $2E	27 41 (#) 44...44 (#) 46
Clear All Tabs	ESC 0	$1B $30	27 48
Place Print Head at # Dot Column ***	ESC F (#)	$1B $46 (#)	27 70 (#)

(continued)

Table 4.71. Continued

Printer Settings

Command	*Form*	*Hexadecimal*	*Decimal*
No CR before LF or FF	ESC I 1	$1B $6C $31	27 108 49
Insert CR before LF or FF ‡	ESC I 0	$1B $6C $30	27 108 48
Erase Current Line from Buffer	Ctrl-X	$18	24
Reset Defaults	ESC c	$1B $63	27 99
Select Printer **** ‡	Ctrl-Q	$11	17
Deselect Printer ****	Ctrl-S	$13	19
Send ID String	ESC ?	$1B $3F	27 63

Notes:

- •* See 4.70. ImageWriter II Software Switches.
- •** Colors are as follows: 0 ($30) Black ‡
  1 ($31) Yellow
  2 ($32) Magenta
  3 ($33) Cyan
  4 ($34) Orange (Yellow+Magenta)
  5 ($35) Green (Yellow+Cyan)
  6 ($36) Purple (Magenta+Cyan)
- •*** See 4.72. ImageWriter Character Pitch Effects on Functions.
- •**** Not available on ImageWriter LQ (II only).
- •† Ignore/Include 8th Data Bit Setting not available on ImageWriter LQ; Software Select Enable/Disable not available on ImageWriter LQ.
- •¥ Not available on ImageWriter II (LQ only).
- •§ Select NLQ font on ImageWriter LQ.
- •‡ Indicates default setting.

Source:

ImageWriter II Technical Reference Manual, pages 39 to 123
ImageWriter LQ Technical Reference Manual, pages 225 to 227

See Also:

4.70. ImageWriter II Software Switches
4.72. ImageWriter Character Pitch Effects on Functions

4.72. ImageWriter Character Pitch Effects on Functions

Char. Pitch	*Left Margin Range*	*Tab Setting Range*	*Max. Dots per Line*	*Horiz. Res. Range*
9 cpi—Extended	0 to 71	1 to 72	576	8 to 576
10 cpi—Pica	0 to 79	1 to 80	640	8 to 640
12 cpi—Elite	0 to 95	1 to 96	768	8 to 768
13.4 cpi—Semicondensed	0 to 106	1 to 107	856	8 to 856
15 cpi—Condensed	0 to 119	1 to 120	960	8 to 960
17 cpi—Ultracondensed	0 to 135	1 to 136	1088	8 to 1088
144 dpi—Proportional Pica	0 to 71	1 to 72	1152	8 to 1152
160 dpi—Proportional Elite	0 to 79	1 to 80	1280	8 to 1280

Notes: Dots per Line and Horizontal Resolution Range apply to printing bit images.

Source: ImageWriter II Technical Reference Manual, pages 60, 66, 106 to 107

See Also: 4.71. ImageWriter Printer Control Functions

4.73. ImageWriter II Custom Character Width Codes

Width (in Dots)	*Width Code (Top 8)*	*Width Code (Bot. 8)*
1	$41	$61
2	$42	$62
3	$43	$63
4	$44	$64
5	$45	$65
6	$46	$66
7	$47	$67
8	$48	$68
9	$49	$69
10	$4A	$6A
11	$4B	$6B
12	$4C	$6C
13	$4D	$6D
14	$4E	$6E
15	$4F	$6F
16	$50	$70

Notes: Width code is part of the Load Custom Characters command (ESC - or ESC +).

Source: ImageWriter II Technical Reference Manual, pages 96, 97

See Also: 4.71. ImageWriter Printer Control Functions
4.74. ImageWriter II Custom Character Data Byte

4.74. ImageWriter II Custom Character Data Byte

*Bit**	*Binary*	*Decimal*	*Hex*
0	0000 0001	1	$01
1	0000 0010	2	$02
2	0000 0100	4	$04
3	0000 1000	8	$08
4	0001 0000	16	$10
5	0010 0000	32	$20
6	0100 0000	64	$40
7	1000 0000	128	$80

Notes: * Bit number corresponds to print head row (0 is top of character).

Source: ImageWriter II Technical Reference Manual, page 93

See Also: 4.71. ImageWriter Printer Control Functions
4.73. ImageWriter II Custom Character Width Codes

4.75. ImageWriter DIP Switch Settings

Switch	*Setting **	*ImageWriter I*	*ImageWriter II*	*ImageWriter LQ*
SW1-1 to SW1-3	O O O	American**	American**	American**
	X O O	Italian	Italian	Italian
	O X O	American	Danish	Danish
	X X O	British	British	British
	O O X	German	German	German
	X O X	Swedish	Swedish	Swedish
	O X X	French	French	French
	X X X	Spanish	Spanish	Spanish
SW1-4	O	66 lines per page**	11" form length**	11" form length**
	X	72 lines per page	12" form length	12" form length
SW1-5	O	Recognizes 8th data bit**	No perforation skip**	No perforation skip**
	X	Ignores 8th data bit	Skip on page perforation	Skip on page perforation
SW1-6 to SW1-7	O O	10 cpi (Pica)	10 cpi (Pica)	10 cpi (Pica)
	X O	12 cpi (Elite)**	12 cpi (Elite)**	12 cpi (Elite)**
	O X	17 cpi (Ultracondensed)	17 cpi (Ultracondensed)	17 cpi (Ultracondensed)
	X X	160 dpi (Elite proportional)	160 dpi (Elite proportional)	Proportional-1 (Elite)
SW1-8	O	No line feed after carriage return**	No line feed after carriage return**	No line feed after carriage return**
	X	Line feed after carriage return	Line feed after carriage return	Line feed after carriage return
SW2-1 to SW2-2	O O	300 baud	300 baud	19,200 baud**
	X O	1200 baud	1200 baud	1200 baud
	O X	2400 baud	2400 baud	2400 baud
	X X	9600 baud**	9600 baud**	9600 baud
SW2-3	O	Data Terminal Ready handshake**	Hardware handshake**	Hardware handshake**
	X	XON/XOFF	XON/XOFF	XON/XOFF
SW2-4	O	NA	Option Card Disabled**	Option Card Disabled**
	X		Option Card Installed	Option Card Installed
SW2-5 to SW2-7	O O O	NA	Hammer Timing Settings***	CSF Bin 1 and Envelope**
	O O X			CSF Bin 1
	X X O			CSF Bins 1,2,3 and Envelope
	X O X			CSF Bins 1 and 2
	X O O			CSF Bins 1,2 and Envelope
	X X X			CSF Bins 1, 2 and 3
SW2-8	O	NA	NA	Auto Paper to Print Line**
	X			Auto Paper to Paper Bail
SW3-1 to SW3-5	-	NA	NA	Hammer Timing Settings***
SW3-6 to SW3-8	-	NA	NA	Vertical Alignment Settings***

Notes:

- NA = not applicable, NS = not specified in manual.
- * O = switch open, X = switch closed.
- ** Indicates factory setting.
- *** Set at factory—do not alter.

Source:

Apple ImageWriter User's Manual, Chapter 4, pages 39, 41, 44, 81 to 82
ImageWriter II Technical Reference Manual, pages 22 to 23
ImageWriter LQ Technical Reference Manual, pages 27 to 28

See Also:

4.69. ImageWriter Default Settings
4.70. ImageWriter II Software Switches

4.76. ImageWriter Alternate Character Sets

	Hexadecimal Character Value											
Language Set	*$23*	*$24*	*$40*	*$5B*	*$5C*	*$5D*	*$5E*	*$60*	*$7B*	*$7C*	*$7D*	*$7E*
American	#	$	@	[	\	]	^	`	{	|	}	~
British	£	$	@	[	\	]	^	`	{	|	}	~
German	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
French	£	$	à	°	ç	§	^	`	é	ù	è	¨
Swedish	#	$	@	Ä	Ö	Å	^	`	ä	ö	å	~
Italian	£	$	§	°	ç	é	^	ù	à	ò	è	ì
Danish	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
Spanish	£	$	§	¡	Ñ	¿	^	`	°	ñ	ç	~

Notes: See 4.75. ImageWriter DIP Switch Settings for information on how to change character set.

Source: ImageWriter II Technical Reference Manual, page 24

See Also: 4.70. ImageWriter II Software Switches
4.75. ImageWriter DIP Switch Settings

4.77. LaserWriter Family History

	Original	*Plus*	*II NT*	*II NTX*
Width	18.5" (469.9mm)	18.5" (469.9mm)	20" (508mm) *	20" (508mm) *
Depth	16.2" (411.5mm) *	16.2" (411.5mm) *	18.9" (480.1mm)	18.9" (480.1mm)
Height	11.5" (292.1mm)	11.5" (292.1mm)	8.5" (215.9mm)	8.5" (215.9mm)
Weight	77 lbs (34.7kg)	77 lbs (34.7kg)	46 lbs (20.7kg)	46 lbs (20.7kg)
CPU	11.16Mhz 68000	11.16Mhz 68000	11.16Mhz 68000	16.66Mhz 68020
RAM	1.5MB	1.5MB	2MB	2MB **
ROM	512K	1MB	1MB	1MB ***
Ports	RS-232, AppleTalk	RS-232, AppleTalk	RS-232, AppleTalk	RS-232, AppleTalk, SCSI
Fonts	4 families ****	11 families †	11 families †	11 families †
Software	PostScript, Diablo 630	PostScript, Diablo 630	PostScript, Diablo 630	PostScript, Diablo 630, LaserJet+
Page Capacity	100 in, 20 out	100 in, 20 out	200 in, 100 out	200 in, 100 out
Introduced	Jan-85	Jan-86	Jan-88	Jan-88

Notes:
- * Without paper trays; trays add 12" on Original and Plus, 6.5" on II NT and II NTX.
- ** May be expanded to 12 MB.
- *** May be expanded on expansion board to 8 MB.
- **** Times, Helvetica, Courier, and Symbol.
- † Times, Helvetica, Helvetica Narrow, Courier, Symbol, Palatino, ITC Avant Garde, ITC Bookman, New Century Schoolbook, ITC Zapf Chancery, ITC Zapf Dingbats.

Source: Apple LaserWriter Reference, pages 147 to 150

See Also: 4.68. ImageWriter Family History

4.78. LaserWriter NTX Disk and File-System Operators

Operator	*Syntax **	*Comments*
cartstatus	string CARTSTATUS false string CARTSTATUS type id filetype true	No card, device does not exist Information about device or file referenced by string
deletefile	namestring DELETEFILE	Deletes referenced file (namestring)
devdismount	device DEVDISMOUNT	Dismounts referenced device (device)
devmount	device DEVMOUNT boolean	Mounts referenced device (device); false return indicates device not mounted
devstatus	device DEVSTATUS free size	Returns status of referenced device
diskonline	DISKONLINE boolean	Returns true if device referenced is online and attached
diskstatus	DISKSTATUS free total	Returns information about the disk
initializedisk	pages action INITIALIZEDISK	Formats the disk and initializes the file system
filenameforall	pattern proc scratchstring FILENAMEFORALL	Executes proc for each file found matching pattern; scratchstring passed to proc
renamefile	oldnamestring newnamestring RENAMEFILE	Renames a file
setuserdiskpercent	percent SETUSERDISKPERCENT	Reserves percentage of file system for user files
status	stringname STATUS pages bytes access creation true	Returns information about file referenced in stringname
userdiskpercent	USERDISKPERCENT percent	Returns percentage of file system space for user files

Notes: * In standard PostScript notation.

Source: LaserWriter Reference, pages 46 to 52

4.79 PostScript Page Types

Page Type	*Printable Region*	*Page Size*	*Comments*
Letter	8" x 10.78"	8.5" x 11"	Standard American letter page
Legal	8" x 13.78"	8.5" x 14"	Standard American legal page
A4	7.79" x 11.08"	8.27" x 11.69"	Standard European A4-sized page
B5	6.45" x 9.76"	6.93" x 9.84"	Standard European B5-sized page
Lettersmall	7.68" x 10.16"	8.5" x 11"	Nonstandard American letter-sized page
A4small	7.47" x 10.85"	8.27" x 11.69"	Nonstandard European A4-sized page
Note	Varies with paper tray*	Varies with paper tray*	

Notes: * Selects Lettersmall if letter-sized tray is installed, A4small if A4-sized tray is installed, legal if legal-sized tray is installed, or B5 if B5-sized tray is installed.

Source: Apple LaserWriter Reference, pages 84 to 86

4.80. PostScript Fcnt Numbers

Font Number	*Font Name*	*Font Style*
0	Courier	Normal
1	Courier	Bold
2	Courier	Oblique
3	Courier	BoldOblique
4	Times	Roman
5	Times	Bold
6	Times	Italic
7	Times	BoldItalic
8	Helvetica	Normal
9	Helvetica	Bold
10	Helvetica	Oblique
11	Helvetica	BoldOblique
12	Symbol	Normal
13	ITC Avant Garde	Book
14	ITC Avant Garde	BookOblique
15	ITC Avant Garde	Demi
16	ITC Avant Garde	DemiOblique
17	ITC Bookman	Demi
18	ITC Bookman	DemiItalic
19	ITC Bookman	Light
20	ITC Bookman	LightItalic
21	Helvetica Narrow	Normal
22	Helvetica Narrow	Bold
23	Helvetica Narrow	BoldOblique
24	Helvetica Narrow	Oblique
25	New Century Schoolbook	Roman
26	New Century Schoolbook	Bold
27	New Century Schoolbook	Italic
28	New Century Schoolbook	BoldItalic
29	Palatino	Roman
30	Palatino	Bold
31	Palatino	Italic
32	Palatino	BoldItalic
33	ITC Zapf Chancery	MediumItalic
34	ITC Zapf Dingbats	Normal

Source: Apple LaserWriter Reference, page 89

See Also: 2.72. Font ID Numbers

4.81. LaserWriter statusdict Persistent Parameter Operators

Operator	*Syntax **	*Comments*
pagecount	PAGECOUNT int	Returns number of pages printed (NT, NTX only)
pagestackorder	PAGESTACKORDER boolean	Returns True if second page faces back of first page, otherwise False
setpagestackorder **	boolean SETPAGESTACKORDER	Sets pagestackorder (default = False)
ramsize	RAMSIZE int	Returns number of bytes in NTX
setprintername **	string SETPRINTERNAME	Sets printer name (31 characters or fewer)
printername	string PRINTERNAME string	Sets printer name and returns string that is actually used (may be substring)
setdosysstart **	boolean SETDOSYSSTART	Sets Sys/Start file on startup status for NTX
dosysstart	DOSYSSTART boolean	Returns current setting of Sys/Start file on NTX
setsoftwareiomode **	int SETSOFTWAREIOMODE	Sets one of following modes: 0 = PostScript batch, 1 = PostScript interactive 2 = Diablo 630 emulation, 5 = HP LaserJet+ emulation
softwareiomode	SOFTWAREIOMODE int	Returns printer mode (default = 0, PostScript batch)
sethardwareiomode	int SETHARDWAREIOMODE	Sets on of following modes: 0 = serial, 2 = AppleTalk
hardwareiomode	HARDWAREIOMODE int	Returns hardware IO mode setting (default = 0)
setsccbatch **	int int int SETSCCBATCH	Sets channel, baud, and options (see 4.84.)
sccbatch	int SCCBATCH int int	Returns baud rate and options for hardware channel in first int
setdostartpage **	boolean SETDOSTARTPAGE	Specifies whether NT/NTX is to perform test page on power-on
dostartpage	DOSTARTPAGE boolean	Returns setting of test page status on NT/NTX
setmargins **	int int SETMARGINS	Sets top and left margins for physical page
margins	MARGINS int int	Returns current top and left margin settings
setpagetype **	int SETPAGETYPE	Sets page type (0 = letter, 1 = note)
pagetype	PAGETYPE int	Returns current page type
setdefaulttimeouts **	int int int SETDEFAULTTIMEOUTS	Sets job, manual feed, and wait timeouts simultaneously
defaulttimeouts	DEFAULTTIMEOUTS int int int	Returns job, manual feed, and wait timeouts
setpassword	int int SETPASSWORD boolean	Changes password from first int to second, returns True if change made
checkpassword	int CHECKPASSWORD boolean	Returns True if int is equal to current system-administrator password
setidlefonts **	mark font sx sy rot nchars... SETIDLEFONTS	See source for description of command; sets characters to be scan converted while NTX is idle
idlefonts	IDLEFONTS mark font sx sy rot nchars...	See source for description of command; returns setidlefonts settings
seteescratch **	int int SETEESCRATCH	Writes second int at position determined by first int into LaserWriter scratch EEROM
eescratch	int EESCRATCH int	Returns value at position (first int) from LaserWriter scratch EEROM

Notes:

- * In standard PostScript notation.
- ** Cannot be performed unless you have previously escaped from normal server save/restore context.

Source:

LaserWriter Reference, pages 91 to 101

See Also:

4.78. LaserWriter NTX Disk and File-System Operators
4.84. LaserWriter setsccbatch Options
4.85. LaserWriter statusdict Volatile Parameter Operators
4.86. LaserWriter New PostScript Operators

4.82. LaserWriter setsoftwareiomode Options

Setting Value	*Description*
0	PostScript batch
1	PostScript interactive
2	Diablo 630 emulation
3	NOT USED
4	NOT USED
5	HP LaserJet+ emulation

Notes: New settings do not take effect until the current job is completed.

Source: Apple LaserWriter Reference, page 94

See Also: 4.81. LaserWriter statusdict Persistent Parameter Operators

4.83. LaserWriter sethardwareiomode Options

Setting Value	*Description*
0	Serial
1	NOT USED
2	AppleTalk

Notes: New settings do not take effect until the current job is completed.

Source: Apple LaserWriter Reference, page 95

See Also: 4.81. LaserWriter statusdict Persistent Parameter Operators

4.84. LaserWriter setsccbatch Options

	Bit Numbers								
Field to Set	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*	*Allowable Values*
Parity							x	x	00 = space, 01 = odd, 10 = even, 11 = mark
Flow Control				x	x	x			000 = xon/xoff, 001 = dtr, 010 = etx/ack
Data Bits		x	x						00 = std, 01 = 7 bits, 10 = 8 bits
Stop Bits	x								0 = 1 bit, 1 = 2 bits

Notes: Note that the Apple manual is misleading in that field values are really bit settings shown in decimal.

Source: Apple LaserWriter Reference, page 96

See Also: 4.81. LaserWriter statusdict Persistent Parameter Operators

4.85. LaserWriter statusdict Volatile Parameter Operators

Operator	*Syntax **	*Comments*
setjobtimeout	Int SETJOBTIMEOUT	Int represents a value, in seconds
jobtimeout	JOBTIMEOUT int	Returns timeout value, in seconds (default = 0)
manualfeedtimeout	MANUALFEEDTIMEOUT int	Returns manual feed timeout, in seconds (default = 60)
waittimeout	WAITTIMEOUT int	Returns wait timeout, in seconds (default = 30)
manualfeed	MANUALFEED boolean	Returns setting of manual feed switch (default = FALSE, not manual feeding)
jobname	JOBNAME string	Returns name of current job
printerror	Status tries PRINTERERROR	Status represents error code, tries represents number of retries so far
product	PRODUCT string	Returns name of printer job is running on
revision	REVISION int	Returns revision of the PostScript interpreter (3 in NTX)
appletalktype	APPLETALK string	Returns type portion of printer's AppleTalk name

Notes: * In standard Forth and PostScript notation.

Source: LaserWriter Reference, pages 102 to 105

See Also: 4.78. LaserWriter NTX Disk and File-System Operators
4.81. LaserWriter statusdict Persistent Parameter Operators
4.86. LaserWriter New PostScript Operators

4.86. LaserWriter New PostScript Operators

Operator	*Syntax **	*Comments*
setpacking	Boolean SETPACKING	Sets array-packing mode to boolean value
currentpacking	CURRENTPACKING boolean	Returns current array-packing mode
packedarray	Any0...anyN-1 n PACKEDARRAY packedarray	Creates a packed array object as defined; see source for details
setcacheparams	Int int int SETCACHEPARAMS	Sets topmost mark, lower, and upper values for cache
currentcacheparams	CURRENTCACHEPARAMS int int int	Returns current cache parameters for mark, lower, and upper (default=mark, 1250, 12500)

Notes: * In standard Forth and PostScript notation.

Source: LaserWriter Reference, pages 110 to 112

See Also: 4.78. LaserWriter NTX Disk and File-System Operators
4.81. LaserWriter statusdict Persistent Parameter Operators
4.84. LaserWriter setsccbatch Options
4.85. LaserWriter statusdict Volatile Parameter Operators

4.87. LaserWriter Reserved Communication Codes

Code	*PostScript Interpretation*
Ctrl-C ($03)	Interrupt
Ctrl-D ($04)	End of file
LF ($0A)	End of line (ignored if following CR)
CR ($0D)	End of line
Ctrl-Q ($11)	XON—start output
Ctrl-S ($13)	XOFF—stop output
Ctrl-T ($14)	Status query

Notes: These characters are not passed on in the data stream.

Source: LaserWriter Reference, page 145

4.88. LaserWriter NT Diablo Emulation Supported Control Characters

Diablo Control Character Sequence	*Description*
ESC BS (#)	Backspace (#) spaces
ESC HT (#)	Horizontal tab to column (#)
ESC LF	Reverse line feed
ESC VT (#)	Vertical tab to line (#)
ESC FF (#)	Set lines per page to (#)
ESC CR P	Reset the printer
ESC DC1 (#)	Set proportional spacing offset to (#)
ESC RS (#-1)	Set vertical motion increment to (#)
ESC US (#-1)	Set horizontal motion increment to (#)
ESC &	End bold and shadow printing
ESC '	Begin reverse printing
ESC -	Set vertical tab at present line
ESC .	Begin normal printing
ESC 0	Set right margin to present position
ESC 1	Set horizontal tab to present position
ESC 2	Clear all tabs, horizontal and vertical
ESC 5	Begin printing in forward paper direction
ESC 6	Begin printing in reverse paper direction
ESC 8	Clear horizontal tab at current position
ESC 9	Set left margin to current position
ESC =	Begin center justification of lines
ESC C	Clear margins at top and bottom
ESC L	Set bottom margin to current position
ESC S	Revert to horizontal motion increment set by hardware switches
ESC T	Set top margin to current position
ESC U	Half-line feed (subscript)
ESC D	Reverse half-line feed (superscript)
ESC E	Begin automatic underscore of characters
ESC R	End automatic underscore
ESC O	Begin bold printing of character
ESC W	Begin shadow printing of characters
ESC X	Clear all character printing modes (bold, shadow, etc.)
ESC M	Begin printing with full justification

Notes:
- Capitalized words in code column are ASCII code name.
- Printing is done in Courier 10 pitch with auto line feed OFF.

Source: LaserWriter Reference, page 67

See Also: 4.89. LaserWriter NT Diablo Emulation eescratch Location Assignments

4.89. LaserWriter NT Diablo Emulation eescratch Location Assignments

eescratch Location	*Assignment*	*Comments*
58	Special switch mode	0 = Diablo 630 emulation, 1= interactive mode
59	Diablo auto LF feature	0 = OFF, 1 = ON
60	Character pitch	0 = 10 cpi; other allowable values: 10, 12, 15
61	Font ID for bold font	1 = Courier Bold; 255 = disable Courier Bold
62	Font ID for normal font	0 = Courier
63	USED INTERNALLY	

Source: LaserWriter Reference, page 69

See Also: 4.88. LaserWriter NT Diablo Emulation Supported Control Characters

4.90. LaserWriter NT LaserJet+ Emulation Font Mapping

Font ID	*LaserJet+ Font*	*LaserWriter NT Font Used*
3	Courier	Courier
4	Helv	Helvetica
5	Tms Rmn	Times Roman

Source: LaserWriter Reference, page 74

See Also: 4.91. LaserWriter NT LaserJet+ Emulation Symbol Set Differences
4.92. LaserWriter NT LaserJet+ Emulation Supported Control Characters

4.91. LaserWriter NT LaserJet+ Emulation Symbol Set Differences

Char. Code	*LaserJet+ Character*
$7F	Gray patch (rubout)
$B0	Overline
$B3	Degree symbol
$E3	Stroked uppercase D (Eth)
$E4	Stroked lowercase d (eth)
$F0	Uppercase Thorn
$F1	Lowercase Thorn
$F7	One-fourth symbol
$F8	One-half symbol
$FC	Solid black square
$FE	Plus/minus sign

Notes: All Roman8 symbol characters map to the Adobe Symbol font equivalent, except for those noted above.

Source: LaserWriter Reference, page 75

See Also: 4.90. LaserWriter NT LaserJet+ Emulation Font Mapping
4.92. LaserWriter NT LaserJet+ Emulation Supported Control Characters

4.92. LaserWriter NT LaserJet+ Emulation Supported Control Characters

LaserJet+ Control Char. Sequence	*Description*
Page Orientation	
ESC&l00	Set portrait mode page orientation
ESC&l10	Set landscape mode page orientation
Font Selection	
ESC(8U	Use Roman8 symbol set
ESC(OU	Use USASCII symbol set
ESC(s1P	Use proportional character spacing
ESC(s0P	Used fixed character spacing
ESC(s10H	Set pitch to 10 cpi
ESC(s12H	Set pitch to 12 cpi
ESC(s16.6H	Set pitch to 16.6 cpi
ESC(s7C	Set character size to 7 point type
ESC(s8C	Set character size to 8 point type
ESC(s8.5C	Set character size to 8.5 point type
ESC(s10C	Set character size to 10 point type
ESC(s12C	Set character size to 12 point type
ESC(s14.4C	Set character size to 14.4 point type
ESC(s0S	Use upright character style
ESC(s1S	Use italic character style
ESC(s-3B	Use light stroke weight
ESC(s0B	Use medium stroke weight
ESC(s3B	Use bold stroke weight
ESC(s3T	Set font to Courier
ESC(s0T	Set font to Line Printer (LaserWriter uses Courier at 16.67 pitch)
ESC(s4T	Set font to Helv (LaserWriter uses Helvetica)
ESC(s5T	Set font to Tms Rmn (LaserWriter uses Times Roman)
ESC(s6T	Set font to Gothic
ESC(s8T	Set font to Prestige Elite
Page Length	
ESC&1#P	Set lines per page to #
ESC&1#E	Set lines for top margin to #
ESC&1#F	Set lines for bottom margin to #
Margins	
ESC9	Clear margin
ESC&a#L	Set left margin at column #
ESC&a#M	Set right margin at column #
Vertical Line Spacing	
ESC&1#C	Line increments in #/48ths of an inch
ESC&l1D	Set printing to 1 line per inch
ESC&l2D	Set printing to 2 lines per inch
ESC&l3D	Set printing to 3 lines per inch
ESC&l4D	Set printing to 4 lines per inch
ESC&l6D	Set printing to 6 lines per inch
ESC&l8D	Set printing to 8 lines per inch
ESC&l12D	Set printing to 12 lines per inch
ESC&l14D	Set printing to 14 lines per inch
ESC&l16D	Set printing to 16 lines per inch
ESC&l24D	Set printing to 24 lines per inch
ESC=	Perform half-line feed

(continued)

Table 4.92. Continued

LaserJet+ Control Char. Sequence	*Description*
Raster Graphics	
ESC*t75R	Set graphics to 75 dpi
ESC*t100R	Set graphics to 100 dpi
ESC*t150R	Set graphics to 150 dpi
ESC*t300R	Set graphics to 300 dpi
ESC*r0A	Start graphics at leftmost position
ESC*r1A	Start graphics at current position
ESC*b#W (data)	Raster graphics data (# items)
ESC*rB	End raster graphics
Cursor Positioning	
ESC&a#R	Set cursor to row #
ESC&a#C	Set cursor to column #
ESC&a#H	Set # of horizontal decipoints
ESC&a#V	Set # of vertical decipoints
Underline	
ESC&dD	Begin underlining characters
ESC&d@	End underlining characters
Display Functions	
ESCY	Turn display functions on
ESCZ	Turn display functions off
ESC&p#X (data)	Print transparent # of bytes of data
Perforation Skip	
ESC&l0L	Disable perforation skip
ESC&l1L	Enable perforation skip
Miscellaneous Features	
ESC&k#H	Set horizontal motion index to #/120ths of an inch
ESC&k0S	Set standard font pitch
ESC&k2S	Set compressed font pitch
Line Termination	
ESC&k0G	CR = CR, LF = LF, FF = FF
ESC&k1G	CR = CR + LF, LF = LF, FF = FF
ESC&k2G	CR = CR, LF = CR + LF, FF = CR + FF
ESC&k3G	CR = CR + LF, LF = CR + LF, FF = CR + FF
ESC&s0C	Enable end-of-line wrapping
ESC&s1C	Disable end-of-line wrapping
ESC&l#X	Select # of copies
Paper Input Control	
ESC&l0H	Eject page
ESC&l1H	Feed from tray
ESC&l2H	Feed pages manually
ESC&l3H	Feed envelope
Cursor Position	
ESC*p#X	Set horizontal cursor position at # of dots
ESC*p#Y	Set vertical cursor position at # of dots

(continued)

Table 4.92. Continued

Font Management

LaserJet+ Control Char. Sequence	*Description*
ESC*c#D	Set font to font ID #
ESC*c#E	Character code ASCII code #
ESC*c0F	Delete all downloaded fonts
ESC*c1F	Delete all temporary fonts
ESC*c2F	Delete last font ID specified
ESC*c3F	Delete last font ID and character code specified
ESC*c4F	Make temporary font
ESC*c5F	Make permanent font
ESC*c6F	Copy or assign font
ESC)s#W (data)	Create font header # of bytes
ESC(s#W (data)	Download character # of bytes
ESC(#X	Designate primary download font ID #
ESC)#X	Designate secondary download font ID #
ESC(0@	Primary font value 0
ESC(1@	Primary font value 1
ESC(2@	Primary font value 2
ESC(3@	Primary font value 3
ESC)0@	Secondary font value 0
ESC)1@	Secondary font value 1
ESC)2@	Secondary font value 2
ESC)3@	Secondary font value 3

Macros

ESC&f#Y	Macro ID #
ESC&f0X	Start macro
ESC&f1X	Stop macro
ESC&f2X	Execute macro
ESC&f3X	Call macro
ESC&f4X	Enable overlay macro
ESC&f5X	Disable overlay macro
ESC&f6X	Delete macro
ESC&f7X	Delete all temporary macros
ESC&f8X	Delete macro ID
ESC&f9X	Make temporary macro
ESC&f10X	Make permanent macro
ESC&f0S	Push position
ESC&f1S	Pop position

(continued)

Table 4.92. Continued

Advanced Graphics

LaserJet+ Control Char. Sequence	*Description*
ESC*c#A	Set horizontal rule pattern size to # of dots
ESC*c#H	Set horizontal rule pattern size to # of decipoints
ESC*c#B	Set vertical rule pattern size to # of dots
ESC*c#V	Set vertical rule pattern size to # of decipoints
ESC*c0P	Print rule
ESC*c2P	Print grayscale
ESC*c3P	HP pattern
ESC*c2G	2% gray pattern
ESC*c10G	10% gray pattern
ESC*c15G	15% gray pattern
ESC*c30G	30% gray pattern
ESC*c45G	45% gray pattern
ESC*c70G	70% gray pattern
ESC*c90G	90% gray pattern
ESC*c100G	100% gray pattern
ESC*c1G	Vertical lines pattern
ESC*c2G	Horizontal lines pattern
ESC*c3G	Diagonal lines pattern
ESC*c4G	Diagonal lines pattern
ESC*c5G	Grid pattern
ESC*c6G	Diagonal grid pattern

Notes:
- Capitalized words in code column are ASCII code name.
- # refers to user-supplied number, (data) refers to user-supplied data.

Source: LaserWriter Reference, pages 72 to 82

See Also: 4.90. LaserWriter NT LaserJet+ Emulation Font Mapping
4.91. LaserWriter NT LaserJet+ Emulation Symbol Set Differences

4.93. LaserWriter NT/NTX Operational Mode Switch Settings

LaserWriter II NT Switch Settings

SW1	*SW2*	*Setting*
Up	Up	AppleTalk enabled, RS-232C disabled
Up	Dn	PostScript batch mode, RS-232C/RS-422 enabled at 9600 baud, parity ignored
Dn	Up	Emulation mode, RS-232C enabled at 9600 baud
Dn	Dn	PostScript batch mode, RS-232C/RS-422 enabled at 1200 baud, parity ignored

LaserWriter II NTX Switch Settings

SW1	*SW2*	*Setting*
Up	Up	AppleTalk enabled, RS-232C disabled
Up	Dn	RS-232C/RS-422 enabled at 9600 baud
Dn	Up	RS-232C/RS-422 enabled at 1200 baud
Dn	Dn	RS-232C enabled at 9600 baud

SW3	*SW4*	*Setting*
Up	Up	PostScript batch mode
Up	Dn	PostScript interactive mode
Dn	Up	Emulation mode
Dn	Dn	HP LaserJet+ emulation mode

SW5	*SW6*	*Setting*
Up	Up	XON/XOFF handshake protocol
Up	Dn	DTR handshake protocol (data terminal ready)
Dn	Up	ETX/ACK handshake protocol
Dn	Dn	XON/XOFF handshake protocol

Notes: If AppleTalk is enabled, switches 3 through 6 have no effect on NTX.

Source: LaserWriter Reference, page 56

See Also: 4.84. LaserWriter setsccbatch Options

Bibliography

The following works were used as primary or secondary sources of information during the compilation of this book. These works formed my desktop reference group, and I recommend them as the solid core of a reference library for anyone dealing seriously with the Macintosh family. Other valuable works also exist, and I consulted some of them during the course of my research. But on a desert island doing Macintosh development, these would be my first choices.

Adobe Systems, Inc. *PostScript Language Reference Manual.* Reading, MA: Addison-Wesley Publishing Company, 1985.

The official definition of the PostScript language. Required reference for PostScript users.
(ISBN 0-201-10174-2)

Adobe Systems, Inc. *PostScript Language Tutorial and Cookbook.* Reading, MA: Addison-Wesley Publishing Company, 1985.

PostScript is actually a variant of Forth and, as such, a difficult language for first-timers to understand. This book presents a number of programming examples, showing their results and describing subtle idiosyncrasies or problems. Absolutely required if you're going to program PostScript code directly.
(ISBN 0-201-10179-3)

Apple Computer, Inc. *Apple Numerics Manual.* Reading, MA: Addison-Wesley Publishing Company, 1986.

The description of Apple's SANE (floating-point numeric environment). Not particularly essential unless you want to know what's happening in the background, since most compilers directly generate SANE code when you use floating-point numbers.
(ISBN 0-201-17741-2)

Apple Computer, Inc. *Designing Cards and Drivers for the Macintosh II and Macintosh SE.* Reading, MA: Addison-Wesley Publishing Company, 1987.

Just what it says it is. You'll probably also want to get a NuBus reference from IEEE or Texas Instruments, but this will cover the basics you need to know.
(ISBN 0-201-19256)

Apple Computer, Inc. *HyperCard Script Language.* Reading, MA: Addison-Wesley Publishing Company, 1988.

The official description of the HyperCard language. Complete, accurate, and reasonably organized, but you might also try Danny Goodman's two works on HyperCard or Michael Swaine's *Dr. Dobb's Essential HyperCard Handbook* for more illuminating information and better examples.
(ISBN 0-201-17632-7)

Apple Computer, Inc. *ImageWriter II Technical Reference Manual.* Reading, MA: Addison-Wesley Publishing Company, 1986.

The manual that should have come with the printer. Even so, the examples are mostly trivial, and the information is not as organized as it should be. Note that the information is not specific to Macs—Apple II information is also included here.
(ISBN 0-201-17766-8)

Apple Computer, Inc. *ImageWriter LQ Reference.* Reading, MA: Addison-Wesley Publishing Company, 1988.

The LQ is a wider-bed ImageWriter II, with only a few other minor differences. Thus, the majority of this manual is the same as the *ImageWriter II Technical Reference Manual.*
(ISBN 0-201-17751-X)

Apple Computer, Inc. *Imagewriter User's Manual.* Cupertino, CA: Apple Computer, 1983.

The original Imagewriter (note the lowercase w, since made uppercase) was really not much more than a repackaged Toyo Electric OEM printer. This manual describes its features. Note, however, that subsequent ImageWriter models add significant new features and deviate from the information described here. The information here is not specific to Macs—Apple II is also covered.
(Apple #A9L030)

Apple Computer, Inc. *Inside Macintosh,* Volume I. Reading, MA: Addison-Wesley Publishing Company, 1985.

Essential work. This volume contains the original material on the user interface and Interface Toolbox managers. Additional material covers the use of assembly language and a road map to the *Inside Macintosh* series. Note: portions are out of date in reference to Plus, SE, and II models.
(ISBN 0-201-17731-5)

Apple Computer, Inc. *Inside Macintosh,* Volume II. Reading, MA: Addison-Wesley Publishing Company, 1985.

Essential work. This volume contains the original material on the operating system calls. Note: portions are out of date in reference to Plus, SE, and II models. Also, the HFS disk format is not represented here.
(ISBN 0-201-17732-3)

Apple Computer, Inc. *Inside Macintosh,* Volume III. Reading, MA: Addison-Wesley Publishing Company, 1985.

Originally highly useful as a summary of the first two volumes. Also includes some basic hardware information. But with subsequent models having

been released, this summary and index is out of date and nonessential.
(ISBN 0-201-17733-1)

Apple Computer, Inc. *Inside Macintosh*, Volume IV. Reading, MA: Addison-Wesley Publishing Company, 1986.

Essential work. Introduces the differences between the Plus and earlier models and also describes HFS format for first time. Other significant additions include the List Manager and the SCSI Manager.
(ISBN 0-201-05409-4)

Apple Computer, Inc. *Inside Macintosh*, Volume V. Reading, MA: Addison-Wesley Publishing Company, 1988.

Essential work. Introduces the differences between the SE and II and earlier models; also describes color QuickDraw for the first time. Other significant additions include the Script Manager and the Shutdown Manager.
(ISBN 0-201-17719-6)

Apple Computer, Inc. *Inside Macintosh X-REF*. Reading, MA: Addison-Wesley Publishing Company, 1988.

Essential work. A complete cross-reference and index of the five earlier volumes, it really is the only way (other than this book) to find anything quickly in the Apple literature.
(ISBN 0-201-13694-5)

Apple Computer, Inc. *LaserWriter Reference*. Reading, MA: Addison-Wesley Publishing Company, 1988.

A very strange work. It includes bits and pieces you won't find anywhere else (like the LaserWriter II NT and II NTX extensions to PostScript), but it doesn't include complete information about anything. Organization, as is typical with Apple books, is hodge-podge at best. What's here is useful, but you'll look in vain for many everyday pieces of data.
(ISBN 0-201-19258-6)

Apple Computer, Inc. *Macintosh Family Hardware Reference*. Reading, MA: Addison-Wesley Publishing Company, 1988.

The first official description of the Macintosh hardware by Apple. The book has many important descriptions of hardware details that don't appear elsewhere (as in the ADB section), but is nevertheless woefully incomplete, highly repetitive, and badly organized.
(ISBN 0-201-19255-1)

Apple Computer, Inc. *Macintosh Technical Notes*, 1–211. Cupertino, CA: Apple Computer, 1985–1987, 1988, 1989.

Absolutely essential. These notes are freely distributed by Apple and disseminated through other organizations. They contain corrections, illuminations, interpretations, and deviations from the *Inside Macintosh* series and, on occasion, recommendations about how to organize or code a program.
(Apple #s M7030, M7032, M0058LL/A)

Apple Computer, Inc. *Programmer's Introduction to the Macintosh Family*. Reading, MA: Addison-Wesley Publishing Company, 1988.

A simplified, well-organized discussion of much of the *Inside Macintosh* information. A good starting point for programmers new to the Mac, but not much use to someone already familiar with the machine.
(ISBN 0-201-19254-3)

Apple Computer, Inc. *Technical Introduction to the Macintosh Family*. Reading, MA: Addison-Wesley Publishing Company, 1987.

Contains much overlap with the *Programmer's Introduction to the Macintosh Family*, but includes a hardware summary not in the other volume. Again, a good starting place for the uninitiated, but not something that will be referred to over and over.
(ISBN 0-201-17765-X)

Chernicoff, Stephen. *Macintosh Revealed*, Volumes 1 and 2. 2nd ed. Indianapolis, IN: Hayden Books, 1987.

Probably the first entry-level work (originally published in 1985) to explain the strange world of Mac programming. The first volume, *Unlocking the Toolbox*, provides an overview, telling you what the various managers do and how they're organized. The second volume, *Programming with the Toolbox*, builds a sample application to show exactly how things work. The second edition has been explicitly updated with information about HFS and the Plus and SE models.
(ISBN 0-672-48400-5 Volume 1)
(ISBN 0-672-48401-3 Volume 2)

Knaster, Scott. *How to Write Macintosh Software*. Hasbrouck Heights, NJ: Hayden Book Company, 1986.

An oddly titled book, since much of the book really deals with how to debug and fix software. Scott was one of the preeminent support specialists accessible to developers during the early Mac days, and he learned what problems most developers ran into; this book is basically his summary of those problems and how to get around them. Essential reading for newcomers.
(ISBN 0-8104-6564-7)

Knaster, Scott. *Macintosh Programming Secrets*. Reading, MA: Addison-Wesley Publishing Company, 1988.

All kinds of wonderful and useful information about the newer versions of Apple's computers and software, with specific, excellent advice about color, the Printing Manager, and the Window Manager as well

as hints on making your software more responsive to users. Good reading for advanced programmers. (ISBN 0-201-06661-0)

Mathews, Keith, and Friedland, Jay. *Encyclopedia Mac ROM*. New York: Brady, 1988.

An encyclopedia of all the Macintosh trap calls, in alphabetical order, with short definitions and examples of how to call the routine in Pascal, C, and assembly language. Unfortunately, the C examples are in MPW C, with strings passed in Pascal form. This doesn't translate very well to the most popular programming language for the Mac, Lightspeed C. Still, a handy book for assembly-language programmers, and, to a lesser degree, C programmers. (ISBN 0-13-541509-8)

Smith, David E., ed. *MacTutor* reprints, Volumes 1–4. Anaheim, CA: MacTutor, 1986, 1987, 1988, 1989.

Reprints of the articles that have appeared in this, the only Macintosh programming journal. *Inside Macintosh* and*The Programmer's Apple Mac Sourcebook* provide the raw data, but to see the data structures and traps in actual use, you'll want to take a look at Volume 1, *The Best of MacTutor*; Volume 2, *The Complete MacTutor*; Volume 3, *The Essential MacTutor*; and Volume 4, *The Definitive MacTutor*. Every Mac programmer I know cribs from this magazine, so don't be left out!
(ISBN 0-9617544-0-0 Volume 1)
(ISBN 0-9617544-1-9 Volume 2)
(ISBN 0-9617544-2-7 Volume 3)
(ISBN 0-9617544-3-5 Volume 4)

Index

Numbers

68000, 68020, 68881, 68851PMU, Z8530 ROMs 392–402
68000 pinouts 407–8
68020 ROM
  long short bus cycle fault stack frames 400
  module call stack frame 401
  pinouts 409
680x0 ROM 392–402
  addressing modes 396
  address space types 398
  coprocessor interface register set map 402
  data formats 13
  exception stack frames 399
  exception vectors 397
  instruction set 393–95
  normal stack frames 398
  register set 392
  special status word 401
  stack frames 398–402
68851 pinouts 411
68881 pinouts 410

A

ABusRecord data structure, AppleTalk Manager 359
actb (alert color table) resource formats 35
ADB extended key codes 383
ADB Manager
  ADBDataBlock data structure 346
  ADBSet InfoBlock data structure 347
  bus command formats 346
  bus register 3 data fields 346
  bus routines 345
  command formats 346
ADB standard key codes 382
AddrBlock data structure, AppleTalk Manager 362
addressing modes, 680x0 396
address space types, 680x0 398
advanced control calls
  disk driver 316
  serial driver 323
ALAP frame, AppleTalk Manager 361
alert(s)
  actb (alert color table) resource format 33
  ALRT (alert template) resource format 35
  DITL resource format 41
  standard icons 31
Alert StageList data structure, Dialog Manager 252
AlertTemplate data structure, Dialog Manager 252
allocation map, MFS 101
ALRT (alert template) resource format 35
AppFile data structure, Segment Loader 280
Apple II, memory organization 276
Apple DeskTop Bus pinouts 413
Apple Macintosh. *See* Macintosh hardware
Apple modem serial port connectors 421
Apple SE, memory organization 275
AppleTalk Manager 353–66
  ABusRecord data structure 359
  AddrBlock data structure 362
  ALAP frame 361
  ATP Client Internet Address data structure 364
  ATP flags field 364
  ATP Packet data structure 364
  ATPParamBlock data structure 363
  DDP Datagram data structure 362
  EntityName data structure 365
  MPParamBlock data structure 360
  names table entry data structure 365
  NBP Packet data structure 365
  result codes 366
  RetransType data structure 366
  routines 353–58
  XPPParamBlock data structure 361
Apple XL, memory organization 275
ASCII
  character set 19
  control codes 18
  key mapping (software) resource format 47
ATP Client Internet Address Data Structure, AppleTalk Manager 364
ATP Flags Field, AppleTalk Manager 364
ATP Packet data structure, AppleTalk Manager 364
ATPParamBlock data structure, AppleTalk Manager 363
attributes, resource 33
  Resource Manager 194
AuxCtlRec data structure 238
AuxWinRec data structure, Window Manager 233

B

B*-Tree Node record layout 103
binary coded decimal number format 17
binary-decimal package routines, Package Manager 260
binary numbers
  conversion to signed decimal numbers 9
  decimal numbers converted to 10
  digit positions 22
  hexadecimal and decimal conversions of 8
  hexadecimal numbers converted to 4
  two's complements 12
BNDL (bundle) resource format 36
boot blocks layout, HFS 103
buses, pinouts and connectors 413–16

C

catalog key record layout 104
cctb (control color table) resource format 36

CDEF (control definition) resource format 37
cdev (control panel device code) resource type
  message values 37
  resource format 37
  result codes 38
character sets
  ASCII 18, 19
  Macintosh 20–21
  special 118
CharByte and CharType values, Script Manager 225
chooser device ID flags, Device Manager 302
cicn (color icon) Resource Format 38
CInfoPBRec data structure, File Manager 292
clock register commands, real time 388
clut (color lookup table) resource format 39
CMovePBRec data structure, File Manager 293
CMYColor data structure, Color Picker package 350
CNTL (control template) resource format 39
CODE (code segment) resource format 39
ColorInfo data structure, Palette Manager 268
color lookup table (clut) resource format 39
Color Manager 215–16
  CProcRec data structure 216
  ReqListRec data structure 216
  routines 215
  SPRocRec data structure 216
color palette (pltt) resource format 51
Color Picker package
  CMYColor data structure 350
  HSLColor data structure 350
  HSVColor data structure 350
  routines 349
Color QuickDraw 208–14
  additive color table 214
  arithmetic drawing modes 212
  ColorGrafPort data structure 210
  ColorSpec data structure 209
  color table data structure 209
  cursor data structure 213
  GrafVars data structure 213
  icon data structure 213
  PixMap data structure 211
  PixPat data structure 211
  RGBColor data structure 209
  RGB equivalents of old QuickDraw colors 214
  routines 208
  transfer modes 212
color table(s)
  control part identifiers for 239
  dialog 251
  menu 243
  resource formats 35, 36, 39, 49, 56
  window part identifiers for 234
command(s)
  HyperTalk 109–11
  Macintosh Programmer's Workshop 115–20
  MacsBug 122–23
  Sound Manager 321
comparison effect on 68000 status bits, SANE 371
compiler formats, REZ 60–65
connectors. *See* pinouts and connectors
constants, printing control 331
constants and data structures (QuickDraw) 200–207
Control Manager 236–39
  AuxCtlRec data structure 238
  control axis parameter values 239
  control definition messages 239
  ControlRecord data structure 237
  CtlCTab data structure 238
  part codes 237
  part identifiers for color tables 238
  routines 236
  types 237
CProcRec data structure, Color Manager 216
crsr (color cursor) resource format 40
CtlCTab data structure, Control Manager 238
cursor(s)
  Color QuickDraw 213
  crsr resource format 40
  CURS resource format 40
  data masks 41
  predefined 29
  QuickDraw 198, 201
CURS resource format 40–41
CWindowRecord data structure, Window Manager 233

D

DA(s)
  device driver control routine messages 254
  device driver format 254
  standard 30
data formats 12–17
  680x0 13
  binary coded decimal number format 17
  Macintosh 14–15
  memory area terminology 16
  string formats 16
data types, PICT 93
DateTimeRec data structure, operating system utilities 327
dctb (dialog color table) resource format 35
DCtlEntry data structure Device Manager 300
DDP Datagram data structure, AppleTalk Manager 362
decimal numbers
  binary coded format for 17
  binary-decimal package routines 260
  binary numbers converted to signed 9
  conversion to binary 10
  conversion to hexadecimal 11
  digit positions 22
  hexadecimal numbers converted to 3
Deferred Task Manager
  DeferredTask data structure 368
  routines 368
DefOSRec data structure, Start Manager 342

DefStartRec data structure, Start Manager 342
DefVideoRec data structure, Start Manager 342
Desk Manager 253–54
  DA device driver control routine messages 254
  DA device driver format 254
  editCmd parameters 254
  routines 253
device driver(s)
  DA
    control routine messages, Desk Manager 254
    format, Desk Manager 254
  disk (*see* disk driver)
  management (*see* Device Manager)
  standard 30
Device Manager 297–314
  chooser device ID flags 302
  DCtlEntry data structure 300
  device access modes 301
  device driver flags 299
  device driver structure 299
  device ParamBlockRec data structure 298
  device positioning modes 302
  device unit table 301
  graphic devices
    attribute types 307
    gDevice data structure 306
    routines 305
  interrupts
    external/status interrupt vector table 304
    primary interrupt vector table 303
    priority levels 302
    processor priority levels 303
    secondary interrupt vector tables 303, 304
  I/O queue record data structure 300
  result codes 298
  routines 297
  sound devices
    FFSynthRec data structure 314
    FTSoundRec data structure 313
    FTSynthRec data structure 313
    note values 308–12
    routines 307
    SWSynthRec data structure 308
    synthesizer types 308
    tone data structure 313
  standard unit assignments 301
dialog color table 251
  dctb resource format 35
Dialog Manager 249–52
  AlertStageList data structure 252
  AlertTemplate data structure 252
  color table 251
  DialogRecord data structure 251
  DialogTemplate data structure 252
  item lists 250
  item types 250
  routines 249
  standard item constants 251
DialogRecord data structure, Dialog Manager 251
Dialog (DITL) resource format 41
Dialog Template data structure, Dialog Manager 252
Dialog Template (DLOG) resource format 42
digit positions in common bases 22
DInfo data structure, File Manager 286
directory record layout 105
disk driver
  advanced control calls 316
  driver positioning modes 314
  DrvSts data structure 315
  result codes 315
  routines 314
disk format
  HFS 101–6
  Macintosh 90
  MFS 100–101
Disk Initialization package 367
  result codes 367
  routines 367
  routine selectors 367
disks
  SCSI address assignments 391
  state control lines and registers 391
DITL (dialog or alert item list) resource format 41
DLOG (dialog template) resource format 42
Drive Queue (DrvQE1) data structure, File Manager 296
Driver positioning modes, disk driver 314
Driver reset word, serial driver 322
DrvSts data structure, device driver 315
DXInfo data structure, File Manager 287

E

editCmd parameters, Desk Manager 254
Editing selection operators in commands 119–20
emulation, LaserWriter 440–45
EntityName data structure, AppleTalk Manager 365
environment word, SANE 372
equal-tempered scale, sound driver 311–12
error messages, system 88
Event Manager
  event codes 283
  event masks, 284
  event modifier flags 284
  EventRecord data structure 282
  EvQE1 data structure 282
  keyboard event message 283
  KeyTrans data structure 283
  postevent result codes 284
  routines 281
Event Manager, Toolbox. *See* Toolbox Event Manager
EventRecord data structure, Event Manager 282
EvQE1 data structure, Event Manager 282
exception halt stack frame, SANE 373
exception vectors 390
  680x0 397

expressions in Macintosh Programmer's Workshop commands 119
extents key record layout 104

F

FamRec data structure, Font Manager 221
FCBPBRec data structure, File Manager 294
fctb (font color table) resource format 42
FFSynthRec data structure, Device Manager Sound Driver 314
FHeaderRec data structure, Slot Manager 339
file(s)
  creators and types 80–88
  formats (*see* file format(s))
  management (*see* File Manager)
  record layout 105
  standard package routines 264
  standard predefined constants 266
  standard resource IDs 266
  standard routine selectors 265
  supplied with system 78–79
file access modes, File Manager 289
file access permissions, File Manager 289
file control block data structure, File Manager 296
file directory format, MFS 101
file format(s)
  MacBinary 91
  MacPaint 91
  MPW object 123–24
  PackBits 92
  PICT 92
  resource 58
  TIFF 97–99
file information flags, File Manager 288
File Manager 285–96
  CInforPBRec data structure 292
  CMovePBRec data structure 293
  DInfo data structure 286
  drive queue (DrvQE1) data structure 296
  DXInfo data structure 287
  FCBPBRec data structure 294
  file access modes 289
  file access permissions 289
  file control block data structure 296
  file information flags 288
  file positioning modes 288
  FInfo data structure 286
  FXInfo data structure 286
  HParamBlockRec data structure 291
  ParamBlockRec data structure 290
  result codes 287–88
  routines 285
  volume control blocks data structure 294–95
  WDPBRec data structure 293
File Manager result codes 287–88
File Manager routines 285
file positioning modes, File Manager 288
file reference (FREF) resource format 44
files supplied with system 78–79
Finder information data structure, segment loader 279
FInfo data structure, File Manager 286
FixedAndInt data structure, Toolbox utilities 258
FKEY (function key) resource format 42
floppy disk, sectors per track 91
FMInput data structure, Font Manager 219
FMOutput data structure, Font Manager 219
FOND (font family) resource format 43
font characterization table, Font Manager 220
font color table (fctb) resource format 42
font family (FOND) resource format 43
font ID numbers 66–74
Font Manager 217–22
  FamRec data structure 221
  FMInput data structure 219
  FMOutput data structure 219
  font characterization table 220
  font metrics data structure 220
  FontRec data structure 218
  numbers 218
  routines 217
  WidthTable data structure 222
FONT resource format 43
Format 1 snd resource format 54
Format 2 snd resource format 54
Four-Tone Synthesizer 313
Free Form Synthesizer 314
FREF (file reference) resource format 44
FTSoundRec data structure, Device Manager Sound Driver 313
FTSynthRec data structure, Device Manager Sound Driver 313
function key (FKEY) resource format 42
FXInfo data structure, File Manager 286

G

gDevice data structure 306
general information 3–25
GetEnvirons verb parameter values, Script Manager 226
GetScript verb parameter values, Script Manager 226
global variables 189–91
GrafVars data structure, Color QuickDraw 213
graphics devices management 305–7
  attribute Types 307
  gDevice data structure 306
  routines 305

H

hardware 377–446
  68000 family and Z8520 microchips 392–402
  disks 391
  keyboards 378–83
  Macintosh summary 377

hardware *continued*
memory organization 384–88
NuBus 403–7
pinouts and connectors 407–25
printers 426–46
registers and protocols 388–90
serial ports 391
hardware, ImageWriter II 432–33
hardware, LaserWriter 446
hexadecimal numbers 3–7
addition tables 5–6
conversion to binary numbers 4
conversion to decimal numbers 3
decimal numbers converted to 11
digit positions 22
multiplication tables 7
HFS boot blocks layout 103
HFS disk format 101–6
HFS volume information 102
HParamBlockRec data structure, File Manager 291
HSLColor data structure, Color Picker package 350
HSVColor data structure, Color Picker package 350
HyperCard. *See* HyperTalk
HyperTalk
commands 109–11
constants 113
messages 113–14
properties 112

I

ICN# (icon list) resource format 44
icon(s)
cicn (color) resource format 38
Color QuickDraw data structure 213
common user interface 76
ICON resource format 44
standard alert 31
ictb (color table dialog item) resource format 35
IDs
font numbers 66–74
menu select 243
Script Manager 224
Slot Manager sResource Apple-defined 339
Slot Manager Vendorinfo list 340
standard file resource 266
standard package 30
ImageWriter
alternate character sets 433
character pitch effects on functions 430
DB-9 and DB-25 serial port connectors 418
default settings 427
DIP switch settings 432
family history 426
printer control functions 428–30
ImageWriter II 427–33
custom character data byte 431
ImageWriter II *continued*
custom character width codes 431
software switches 428
INIT (initialization) resource format 44
InitZone parameter block, Memory Manager 278
installer
file spec flags 108
resource spec flag 108
script format 107
installer file spec flags 108
installer resource spec flag 108
installer script format 107
instruction set, 680x0 393–95
Int64Bit data structure, Toolbox utilities 258
interface components 76
interface Toolbox Managers. *See* Toolbox Managers
international package routines, Package Manager 261–63
international (INTL) resource formats 45–46
international routine selectors, Package Manager 263
international sort ordering 23–24
international string comparison result codes 263
interrupt priority levels, Device Manager 302
INTL (international) 0 or itl0 resource format 45
Intl0Rec data structure, Package Manager 262
INTL (international) 1 or itl1 resource format 46
Intl1Rec data structure, Package Manager 263
I/O queue record data structure, Device Manager 300
itlb (international utilities package script bundles) resource format 46

J–K

jump table formats, segment loader 280
just-tempered scale, Sound Driver 308–11
KCHR (ASCII key mapping) resource format 47
key(s)
catalog key layout 104
extents key record layout 104
function key resource format 42
key mapping resource formats
KCHR ASCII software 47
KMAP hardware 47
predefined 75
keyboard(s) 378–83
command protocol 378
event message 283
key codes
ADB extended 383
ADB standard 382
international 381
U.S. 380
key transition codes 379
pinouts 424
keyboard/mouse pinouts 424
KeyTrans data structure, Event Manager 283
KMAP (keyboard mapping hardware) resource format 47

L

LaserWriter 434–46
  family history 434
  hardware, NT/NTX operational mode switch settings 446
  software 435–46
    DB-9 serial port connector 419
    DB-25 serial port connector 420–21
    Mini-8 serial port connector 420
    new PostScript operators 439
    NT Diablo emulation eescratch location assignments 441
    NT Diablo emulation supported control characters 440
    NT LaserJet+ emulation font mapping 441
    NT LaserJet+ emulation supported control characters 442–45
    NT LaserJet+ emulation symbol set differences 441
    NTX bus signals (connector) 416
    NTX card layout 416
    NTX disk and file-system operators 435
    PostScript font numbers 436
    PostScript page types 436
    reserved communication codes 440
    sethardwareiomode options 438
    setsccbatch options 438
    setsoftwareiomode options 438
    statusdict persistent parameter operators 437
    statusdict volatile parameter operators 439
Level 1 secondary interrupt vector table, Device Manager 303
Level 2 external/status interrupt vector table, Device Manager 304
Level 2 secondary interrupt vector table, Device Manager 304
List Manager
  ListRec data structure 270
  masks 271
  messages 271
  routines 269
  selection flag byte 270
ListRec data structure, List Manager 270
logical operators
  in commands 119
  truth tables for 25

M

MacBinary file format 91
mach (machine identification) resource format 48
Macintosh hardware
  128/512 DB-9 serial port connectors 417
  128/512/Plus keyboard command protocol 378
  128/512/Plus keyboard pinouts 424
  128/512/Plus key transition codes 379
  128/512/Plus mouse pinouts 424
  128/512/Plus/SE external drive connector 422
Macintosh hardware *continued*
  128/512/Plus/SE general memory organization 386
  128/512/Plus/SE sound buffer 385
  128/512/Plus/SE system RAM layout 384–85
  128/512/Plus/SE video buffer 384
  II general memory organization 387
  II system RAM layout 385
  II video card pinouts 425
  character set 20–21
  data formats 14–15
  disk format 90–106
  key codes
    ADB extended 383
    ADB standard 382
    international 381
    U.S. 380
  machine summary 377
  Plus/Se/II mini-8 serial port connectors 417
  Plus/SE/II SCSI connector 423
Macintosh Programmer's Workshop (MPW)
  command(s) 115–16
    command status values 117
    command terminators 117
    editing selection operators in 119–20
    expressions in commands 119
    special characters in 118
  object file format 123–24
Macintosh Programmer's Workshop (MPW) tools 120–23
  MacsBug (*see* MacsBug)
MacPaint file format 91
MacsBug
  commands 122–23
  exceptions 121
master pointer structure, Memory Manager 277
MBAR (menu bar) resource format 48
MBDF (default menu definition procedure) resource format 48
MCEntry data structure, Menu Manager 242
mctb (menu color table) resource format 49
MDEF (menu definition) resource format 49
memory. *See also* Memory Manager
  system traps/routines that may move or purge 175–88
memory area terminology, 16
Memory Manager
  InitZone parameter block 278
  master pointer structure 277
  memory organization
    for 128/512 Plus 274
    for II 276
    for SE 275
    for XL 275
  result codes 278
  routines 273–74
  tag byte format 277

Memory Manager *continued*
zone block header data structure 277
zone data structure 276
memory organization, hardware 384–88
Macintosh 128/512/Plus SE 386
Macintosh II 387
MenuCRsrc data structure, Menu Manager 243
MenuInfo data structure, Menu Manager 242
Menu Manager 240–44
bar defProc messages 244
definition messages 244
definition metacharacters 244
MCEntry data structure 242
menu color information table 243
MenuCRsrc data structure 243
MenuInfo data structure 242
Menu List data structure 243
MenuSelect ID/item return 243
routines 240–41
MENU resource format 49
MenuSelect ID/item return, Menu Manager 243
MFS disk format 100–101
MFS file directory format 101
MFS volume allocation map 101
MFS volume information 100
microfloppy sectors per track 91
ModifierStub data structure, Sound Manager 320
mouse locations in windows, Window Manager 234
mouse pinouts 424
MPPParamBlock data structure, AppleTalk Manager 360
MPW. *See* Macintosh Programmer's Workshop (MPW)

N

names table entry data structure, AppleTalk Manager 365
NBP Packet data structure, AppleTalk Manager 365
NFNT resource format 50
Not-a-Number codes, SANE 371
note values, Device Manager Sound Driver 308–12
nrct (rectangle positions) resource format 50
NuBus 403–7. *See also* Slot Manager (NuBus)
24-bit to 32-bit memory translation 404–5
block transfer size and address 405
bus signals (connector) 414
ByteLanes mapping 405
card layout 413
Macintosh II pinouts and connectors 413–14
memory usage 403
NuBus to 68020 byte swapping 404
pinouts and connectors 413–14
slot memory assignments 404
TMx acknowledgement codes 406
TMx attention cycle codes
TMx transfer mode codes 406
NullSTRec data structure, TextEdit 247
numeric conversions 3–17

O

opcodes, PICT 94–97
operating system 77–90
error(s) 325
error messages 88
file creators and types 80–88
files supplied with 78–79
global variables 189–91
HyperCard (*see* HyperTalk)
installer 107–8
Macintosh disk format 90–106
Macintosh Programmer's Workshop (*see* Macintosh Programmer's Workshop (MPW))
predefined packages 29
resources (*see* resource(s), operating system)
software history 77
software result codes 89–90, 327
traps (*see* operating system traps)
user interface (*see* user interface)
utilities 326–27
utility routines 326
operating system calls 272
ADB Manager 345–47
AppleTalk Manager 353–66
Color Picker package 349–50
Deferred Task Manager 368
Device Manager 297–304
disk driver 314–16
Disk Initialization package 367
Event Manager 281–84
File Manager 285–96
graphics devices 305–7
Memory Manager 273–78
operating system utilities 326–27
Printing Manager 328–33
SANE 369–73
SCSI Manager 316–18
segment loader 279–80
serial drivers 321–24
Shutdown Manager 343–44
Slot Manager 334–40
sound driver 307–14
Sound Manager 319–21
Start Manager 341–42
system error handler 324–25
time manager 368
vertical retrace manager 351–52
video driver 347–48
operating system traps 128–88
by manager 160–74
memory moved/purged by select 175–88
by name 129–43
by number 144–59
trap word format 128

P

package(s)
  predefined system 29
  standard IDs 30
Package Manager 259–66
  binary-decimal package routines 260
  international package routines 261
  international routine selectors 263
  international string comparison result codes 263
  Intl0Rec data structure 262
  Intl1Rec data structure 263
  routines 259
  SFReply data structure 265
  standard file package routines 264
  standard file resource IDs 266
  standard file routine selectors 265
  supplied PACKs 260
PackBits file format 92
palette data structure, Palette Manager 268
Palette Manager
  Colorinfo data structure 268
  color methods 268
  palette data structure 268
  routines 267
ParamBlockRec data structure, File Manager 290
parameter RAM layout 388
PAT# (pattern list) resource format 51
patches (PTCHs), standard 30
PAT (pattern) resource format 50
PicComments 92–93
PICT data types 93
PICT file format 92–97
PICT opcodes 94–97
PICT (picture) resource format 51
pinouts and connectors 407–25
  68000 pinouts 407–8
  68020 pinouts 409
  68851 pinouts 411
  68881 pinouts 410
  buses 413–16
    LaserWriter NTX 416
    NuBus(II) 413–14
    SE connector 414–15
  keyboard/mouse printouts 424
  Macintosh 128/512/Plus/SE external drive connector 422
  Macintosh II video card pinouts 425
  Macintosh Plus/SE/II SCSI connector 423
  serial port 417–21
  SIMM connector (single inline memory module) 412
  sound pinouts 423
pixel pattern (ppat) resource format 52
PixMap data structure, Color QuickDraw 211
PixPat data structure, Color QuickDraw 211
pltt (color palette) resource format 51
PostEvent result codes 284
PostScript
  font numbers 436
  page types 436
powers of two 22
ppat (pixel pattern) resource format 52
predefined cursors 29
predefined keys 75
predefined system packages 29–32
PRER (Chooser device type) resource format 52
PRES (Chooser device type) resource format 53
primary interrupt vector table, Device Manager 303
printers 426–46
Printing Manager 328–33
  printing control constants 331
  result codes 330
  routines 328
  TDftBitsBlk data structure 333
  TGetRotnBlk data structure 333
  TGnlData data structure 331
  TPrinfo data structure 329
  TPrint data structure 329
  TPrJob data structure 329
  TPrStatus data structure 330
  TRslRec data structure 332
  TRslRg data structure 332
  TSetRslBlk data structure 332
processor priority levels, Device Manager 303
PTCHs. *See* patches (PTCHs), standard

Q

queue types, operating system utilities 327
QuickDraw 195–214. *See also* Color QuickDraw
  bitmap data structure 204
  color (old style) equates 204
  constants 200
  coordinate plane 207
  cursor data structure 201
  cursor routines 198
  FontInfo data structure 206
  grafPort data structure 202
  grafPort routines 197
  InitGraf default settings 207
  miscellaneous routines 197
  pen/line drawing routines 198
  pen state data structure 205
  pen transfer modes 206
  picture data structure 205
  point data structure 203
  polygon data structure 205
  QDProcs data structure 206
  rect data structure 203
  region data structure 203
  regions/pixels/pictures routines 195–96
  text drawing routines 199
  text style (styleitem) equates 202

R

RAM memory layout
  128/512/Plus/SE system 384
  model II system 385
  parameter 388
real time clock register commands 388
record layout
  B*-Tree node 103
  directory 105
  file 105
  thread 106
registers and protocols
  680x0 register set 392
  disk state control lines and registers 391
  exception vectors 390
  real time clock register commands 388
  SCC data and control registers 391
  VIA interrupts 390
  VIA registers 389
  VIA timers 389
register set map, 680x0 coprocessor interface 402
ReqListRec data structure, Color Manager 216
resource(s), operating system 29–74
  attributes 33
  file format 58
  font ID numbers 66–74
  numbering 32
  predefined system packages 29–32
  REZ compiler formats 60–65
  RMAKER syntax 59
  ROM 31–32
  sysEnvirons results 74
  type(s) 33–34
  type formats 35–57
Resource Manager 193–94
  result codes and resource attributes 194
  routines 193
result codes
  AppleTalk 366
  cdev resource type 38
  Device Manager 298
  disk driver 315
  Disk Initialization 367
  Event Manager postevent 284
  File Manager 287–88
  international string comparison 263
  Memory Manager 278
  operating system 89–90, 327
  Printing Manager 330
  Resource Manager 194
  Scrap Manager 256
  SCSI 318
  serial driver 324
  Slot Manager 340
  SysEnv 259
  Vertical Retrace Manager 352
result codes from SANE inquiries, SANE 371
RetransType data structure, AppleTalk Manager 366
REZ compiler formats 60–65
RGBColor data structure, Color QuickDraw 209
RGB equivalents of old QuickDraw colors 214
RMAKER syntax 59
ROM resources 31–32
ROM traps and system calls 127–373
  global variables 189–91
  interface Toolbox Managers 192–271
  operating system calls 272–373
  ROM versions 127
  system traps 128–88
ROM versions 127
routine selectors, international 263
ROv# resource format 53

S

SANE 369–73
  comparison effect on 68000 status bits 371
  data types 370
  environment word 372
  exception halt stack frame 373
  Not-a-Number codes 371
  result codes from SANE inquiries 371
  routines 369
  test-exception word 373
SANE environment word 372
SCC data and control registers 391
Scrap Manager
  data formats 256
  result codes 256
  routines 255
  ScrapStuff data structure 256
Script Manager 223–26
  CharByte and CharType values 225
  GetEnvirons verb parameter values 226
  GetScript verb parameter values 226
  ID numbers 224
  routines 223
  transliterate target values 225
scrn (screen configuration) resource format 53
ScrpSTElement data structure, TextEdit 248
SCSI address assignments 391
SCSI connector, Macintosh Plus/SE/II 423
SCSI Manager
  control and status bits 317
  device partition map 318
  driver descriptor map 318
  operations 317
  result codes 318
  routines 316
SE bus expansion signals (connector) 415
SE connector 414–15
SE expansion card layout 414
segment loader
  AppFile data structure 280
  Finder information data structure 279

segment loader *continued*
jump table formats 280
routines 279
serial drivers
advanced control calls 323
reference numbers 322
reset word 322
result codes 324
routines 321
SerShk data structure 323
SerStaRec data structure 323
serial port pinouts and connectors 417–21
serial ports 391
SerShk data structure, Serial Driver 323
SerStaRec data structure, Serial Driver 323
SetStyle attribute controls, TextEdit 248
SExecBlock data structure, Slot Manager 337
SFReply data structure, Package Manager 265
Shutdown Manager
routines 343
ShutDwnInstall masks 344
SInfoRecord data structure, Slot Manager 338
Single Inline Memory Module (SIMM Connector) 412
SIZE (Switcher/MultiFinder) resource format 54
Slot Manager (NuBus) 334–40. *See also* NuBus
FHeaderRec data structure 339
result codes 340
routines 334–36
SExecBlock data structure 337
SInfoRecord data structure 338
SpBlock data structure 337
SPRAMRecord data structure 339
sResource Apple-defined IDs 339
sResource directory data structure 338
VendorInfo list IDs 340
SndChannel data structure, Sound Manager 320
snth (synthesizer) resource format 55
software, ImageWriter II 428–31
software, LaserWriter 435–45
software history 77
sort ordering, international 23–24
sound buffer, Macintosh 385
sound driver 307–14
FFSynthRecdata structure 314
note values
equal-tempered scale 311–12
just-tempered scale 308–11
routines 307
SWSynthRec data structure 308
synthesizer types 308
Sound Manager
commands 321
ModifierStub data structure 320
routines 319
SndChannel data structure 320
synthesizer types 320
sound pinouts 423
SpBlock data structure, Slot Manager 337
special characters in MPW commands 118
special status word, 680x0 401
sPRAMRecord data structure, Slot Manager 339
SProcRec data structure, Color Manager 216
Square Wave Synthesizer 308–12
sResource Apple-defined IDs, Slot Manager 339
sResource directory data structure, Slot Manager 338
stack frames
68020 long/short bus cycle 400
68020 module call 401
680x0 exception 398
680x0 normal 398
standard alert icons 31
standard drivers and DAs 30
standard file package routines, Package Manager 264–66
standard file predefined constants, Package Manager 266
standard file resource IDs, Package Manager 266
standard file routine selectors, Package Manager 265
standard package IDs 30
standard patches (PTCHs) 30
standard unit assignments, device driver 301
Start Manager
DefOSrec data structure 342
DefStartRec data structure 342
DefVideoRec data structure 342
routines 341
STElement/LHElement data structure, TextEdit 247
STR# (string list) resource format 55
string(s)
formats 16
result codes for international comparison 263
STR resource formats 55
STR (string) resource format 55
StScrpRec data structure, TextEdit 248
supplied PACKs, Package Manager 260
switch settings, LaserWriter 446
SWSynthRec data structure, device driver 308
synthesizer types
sound driver 308
Sound Manager 320
SysEnvirons results 74
SysEnvRec data structure, Toolbox utilities 258
SysEnv result codes, Toolbox utilities 259
system. *See* operating system
system error handler
routine 324
system errors 325
system traps. *See* operating system traps

T

tag byte format, Memory Manager 277
TDftBitsBlk data structure, Printing Manager 333
TERec data structure, TextEdit 246
test-exception word, SANE 373

TEStyleRec data structure, TextEdit 247
TextEdit 245–48
  justification codes 246
  NullSTRec data structure 247
  routines 245
  ScrpSTElement data structure 248
  SetStyle attribute codes 248
  STElement/LHElement data structure 247
  StScrpRec data structure 248
  TERec data structure 246
  TEStyleRec data structure 247
  TextStyle data structure 248
TEXT resource format 55
TGetRotnBlk data structure, Printing Manager 333
TGnlData data structure, Printing Manager 331
thread record layout 106
TIFF file format 97–99
Time Manager
  routines 368
  TMTask data structure 368
TMx acknowledgment codes 406
TMx attention cycle codes 407
TMx line use 406–7
TMx transfer mode codes 406
tone data structure, Device Manager Sound Driver 313
Toolbox Event Manager 227–29
  event codes 228
  event masks 229
  event modifier flags 229
  EventRecord data structure 228
  priorities 227
  routines 227
Toolbox Managers
  Color Manager 215–16
  Color QuickDraw 208–14
  Control Manager 236–39
  Desk Manager 253–54
  Dialog Manager 249–52
  Font Manager 217–22
  List Manager 269–71
  Menu Manager 240–44
  Package Manager 259–66
  Palette Manager 267–68
  QuickDraw 195–214
  Resource Manager 193–94
  Scrap Manager 255–56
  Script Manager 223–26
  TextEdit 245–48
  Toolbox Event Manager 227–29
  Toolbox utilities 257–59
  Window Manager 230–35
Toolbox utilities
  FixedAndint data structure 258
  Int64Bit data structure 258
  routines 257
  SysEnvRec data structure 258
  SysEnv result codes 259
TPrinfo data structure, Printing Manager 329
TPrint data structure, Printing Manager 329
TPrJob data structure, Printing Manager 329
TPrStatus data structure, Printing Manager 330
transliterate target values, Script Manager 225
trap word format 128
TRslRec data structure, Printing Manager 332
TRslRg data structure, Printing Manager 332
truth tables for logical operations 25
TSetRslBlk data structure, Printing Manager 332
two, powers of 22
  two's complements 12

U

user interface 75–76
  common icons 76
  components 76
  predefined keys 75
  Toolbox Managers (*see* Toolbox Managers)
utilities, operating system
  DateTimeRec data structure 327
  queue types 327
  result codes 327
  routines 326
utilities, Toolbox. *See* Toolbox utilities

V

variables, global 189–91
VBLTask data structure, Vertical Retrace Manager 352
VendorInfo list IDs, Slot Manager 340
version
  ROM 127
  software resource format 56
vers (software version number) resource format 56
Vertical Retrace Manager
  result codes 352
  routines 351
  VBLTask data structure 352
VIA interrupts 390
VIA registers 389
VIA timers 389
video buffer, Macintosh 384
video card pinouts 425
video driver (NuBus Card)
  data structures 348
  parameter IDs 347
  VideoDeviceRec data structure 347
volume control blocks data structure, File Manager 294–95
volume information
  HFS 102
  MFS 101

W

wctb (window color table) resource format 56
WDEF (window definition) resource format 57
WDPBRec data structure, File Manager 293

WidthTable data structure, Font Manager 222
Window Manager 230–35
  AuxWinRec data structure 233
  CWindowRecord data structure 233
  mouse locations in windows 234
  routines 230–31
  WinCTab data structure 234
  window definition messages 235
  window hit messages 235
  window part identifiers for color tables 234
  WindowRecord data structure 232
Window Manager *continued*
  window types 232
WIND (window) resource format 57

X–Z

XPPParamBlock data structure, AppleTalk Manager 361
zone block header data structure, Memory Manager 277
zone data structure, Memory Manager 276

Thom Hogan

Thom Hogan is a veteran of the personal-computer industry, with 13 years of experience as a software developer, programmer, technical writer, marketing manager, and lecturer. He is the author of seven books, among them *The Programmer's PC Sourcebook* (Microsoft Press, 1988) and the bestselling *CP/M User's Guide*, which has sold over 300,000 copies worldwide. He is also a frequent contributor to computer magazines, and his columns appear regularly in *Bay Area Computer Currents* and *Macworld*.

Other Titles From Microsoft Press

THE APPLE® MACINTOSH® BOOK, 3rd ed.
Cary Lu

"If you have a Mac or are curious about them, this is the book to have." *BYTE*

The classic book that accompanied the introduction of the Mac in 1984 is back again and thoroughly updated to include information on the Mac II, SE, and Plus. THE APPLE MACINTOSH BOOK is an authoritative and comprehensive examination of the Mac's design philosophy, architecture, hardware and software options, and significant user issues. Cary Lu covers selecting the right Mac, uses of the Mac, internal hardware expansion, mass-storage options, local area networks, and more. Learn about how the Mac works—its video screen, keyboard and mouse, disks and disk drives, data pathways and memory, and printers. You'll find what's currently and soon to be available in word processing, graphics, business software, communications, and desktop publishing for the Mac. Lu provides information about programming languages for the Mac and about how to work with local area networks and AppleTalk. He explores the issue of future directions for the Mac. THE APPLE MACINTOSH BOOK—your Mac library is incomplete without it.
416 pages, softcover *$21.95*
Order Code: 86-96213

EXCEL IN BUSINESS
Number-crunching Power on the Apple® Macintosh®
Douglas Cobb and The Cobb Group

"The definitive guide to Excel on the Macintosh...an outstanding reference, for its readability and detail."
Computer Book Review

If you've been looking for a book that shows how you can harness all the incredible power, speed, and simplicity that Microsoft Excel has to offer, EXCEL IN BUSINESS (now updated to version 1.5) is the perfect source. With step-by-step tutorials and an abundance of working examples, integrated software expert Douglas Cobb walks you through the process of maximizing Excel's potent spreadsheets, creating illustrative charts, turning your spreadsheet into a database manager, and switching on the power of macros. You'll also learn how to customize menus and dialog boxes, take advantage of the new formatting features for spreadsheets and charts, and use each of the 47 powerful new spreadsheet functions. The authors include practical applications that teach you how to produce spreadsheets and graphics, how to work with databases, how to master macros, and how to link Excel to other programs. The book is packed with tutorials and tips.
816 pages, softcover *$24.95*
Order Code: 86-95314

WORKING WITH WORD, 2nd ed.
The Definitive Guide to Microsoft® Word on the Apple® Macintosh®
Chris Kinata and Gordon McComb

When you are ready to go beyond simple word processing with Microsoft Word for the Macintosh, this is *the* book of choice. Now updated for version 4, WORKING WITH WORD is filled with inside advice, detailed information, and tutorials on every Word feature. You'll find scores of excellent tips—many not in the documentation—that add power to your creation of any kind of printed document. Topics cover desktop publishing with Word; integrating graphics into a Word document and wrapping text around them; customizing menus and retrieving lost files; linking Word with other applications; optimizing memory management; creating spreadsheetlike tables; and working with lists and multiple columns. You'll find blueprints for newsletters, multicolumn brochures, correspondence, and reports. WORKING WITH WORD is the best, most complete, and most up-to-date book on Microsoft Word available.
752 pages, softcover *$21.95*
Order Code: 86-96957

MUSIC THROUGH MIDI
Using MIDI to Create Your Own Electronic Music System
Michael Boom

"Mr. Boom is a good teacher, and his book is a sure-footed guide through yet another technological forest."
The New York Times

Turn the promise of MIDI into musical reality with the wealth of up-to-date information in this fact-filled book. If you've never pressed a key on a synthesizer or put a disk in a computer, you'll find inspiration in this introduction to the fundamentals of electronic music, sound generation, and MIDI. You'll find a full explanation of how MIDI sends data back and forth between components, a look at channels and messages, and a pin-by-pin description of MIDI hardware requirements. Tim Gorman offers insights on MIDI in live performance, and Tom Scott gives you the benefit of his experience on MIDI in the recording studio.
304 pages, softcover *$19.95*
Order Code: 86-95884

HYPERTALK™: PROGRAMMER'S QUICK REFERENCE
Lon Poole

HYPERTALK: PROGRAMMER'S QUICK REFERENCE offers every HyperTalk script writer easy access to version 1.2.1 of HyperTalk. This handy guide offers concise, alphabetic referencing to *all* HyperTalk commands, functions, properties, constants, system messages, and operators. Entries are accompanied by short code examples and usage notes, and each shows the context of use as well as syntax.

144 pages, softcover *$5.95*
Order Code: 86-96601

QUICK REFERENCE GUIDE TO MICROSOFT® WORD FOR THE APPLE® MACINTOSH®
Lisa Ann Jacobs

This great little guide offers you a handy action-oriented reference to Microsoft Word through version 4. You can look up specific tasks—customizing menus, indexing, underlining text—without knowing the specific Word commands. It's a practical, fast way to understand and use Microsoft Word. Also included is an index to commands and a summary of keyboard shortcuts.

144 pages, softcover *$5.95*
Order Code: 86-97146

TEMPUS TITLES

THE TOMORROW MAKERS
A Brave New World of Living-Brain Machines
Grant Fjermedal

"Talespinners have nothing on the hard-core science freaks. This nonfiction book has enough new ideas for 16 Star Trek sequels. And better dialogue."
Rudy Rucker, *San Francisco Chronicle*

Award-winning science writer Grant Fjermedal paints a richly human picture of the lives and work of the brilliant and sometimes eccentric robotics researchers at Stanford, MIT, and Carnegie-Mellon. Through Fjermedal's keen journalistic eye, you will see the world of artificial intelligence and robotics and witness the break-neck speeds at which these technologies are evolving. He details the astounding work going on in robotics today, including the process of downloading individual human minds into computerized robots that never die. THE TOMORROW MAKERS is an enthralling portrait of world-class computer scientists and intense hackers who all possess an unmatched passion and ability for "making tomorrow."

288 pages, softcover *$8.95*
Order Code: 86-96247

MACHINERY OF THE MIND
Inside the New Science of Artificial Intelligence
George Johnson

"An ideal presentation of what artificial intelligence is all about."
Douglas Hofstadter, author of *Gödel, Escher, Bach*

Focusing on the work of giants—including Marvin Minsky, Roger Schank, and Edward Feigenbaum—George Johnson gives you an intimate look at the state of AI today. You'll discover how machines are beginning to understand English, discover scientific theories, and create original works of art; how research in AI is helping us understand the human mind; and how AI affects our lives today. If you have any interest at all in science and technology, MACHINERY OF THE MIND is captivating reading.

352 pages, softcover *$9.95*
Order Code: 86-96072

PROGRAMMERS AT WORK
Interviews by Susan Lammers

"PROGRAMMERS AT WORK...does for budding software writers what the Paris Review interviews did for would-be fiction writers. It provides comfort, inspiration and a sense of community with those who have already succeeded in the field....It is not to be missed."
The New York Times

PROGRAMMERS AT WORK is a collection of captivating interviews that probe the minds of twenty of today's most notable programmers. The interviews highlight the forces, the events, and the personality traits that influenced today's software movers and shakers. How do these programmers approach program design? Is programming a talent or a learned skill? An art or a science? And how do these programmers envision the future of computing? Although each interview reveals an individual success story, the collection, taken as a whole, provides a colorful portrait of the microcomputer industry. Included are interviews with Andy Hertzfeld (Macintosh Operating System), John Warnock (PostScript), C. Wayne Ratliff (dBASE), Jonathan Sachs (Lotus 1-2-3), and Bill Gates (BASIC). A lively appendix provides some actual code and worksheets from these software wizards.

400 pages, softcover *$9.95*
Order Code: 86-97195

INVENTORS AT WORK

Interviews by Kenneth A. Brown
Foreword by James Burke

"Highly recommended." *Booklist*

INVENTORS AT WORK is an outstanding collection of 16 engaging and illuminating interviews with the most notable inventors of our time—the creators of the laser, the microprocessor, the human-powered airplane, the implantable pacemaker, the Apple II computer, the plastic soda bottle, and more. You'll take a glimpse into the creative process, look at invention as art and invention by committee, and see the impact of United States patent laws and a capitalist economy on the inventive spirit.

408 pages, softcover *$9.95*
Order Code: 86-96080

OUT OF THE INNER CIRCLE

The True Story of a Computer Intruder Capable of Cracking the Nation's Most Secure Computer Systems

Bill Landreth aka "The Cracker"

OUT OF THE INNER CIRCLE is a highly readable story of adolescent ingenuity and intelligence mixed with high-stakes corporate and government intrigue, written by "The Cracker"—the computer wizard apprehended by the FBI for illegally gaining access to high-level computer systems. "The Cracker"details how he and his "Inner Circle" clique of hackers penetrated the nation's largest and most sensitive computer networks with nothing more than several hundred dollars. OUT OF THE INNER CIRCLE—a first-person look at the secretive hacker subculture and an eye-opening examination of the computer security issue that affects us all.

248 pages, softcover *$9.95*
Order Code: 86-97344

COMPUTER LIB/DREAM MACHINES

Ted Nelson

"An exuberant, multifont compendium of computing proverbs, anecdotes, jokes, predictions, and politics. Still as fresh and relevant as it was a dozen years ago. COMPUTER LIB is a browser's gold mine."
PC World

This is a time whose book has come. First published in 1974, COMPUTER LIB/DREAM MACHINES became the first cult book of the computer generation. Embraced by hackers, quoted by the press, it inspired thousands in the computer industry. Although over ten years old, COMPUTER LIB predicted the major issues of today—design of easy-to-use computer systems, image synthesis, artificial intelligence, computer-assisted instruction, and hypertext—Nelson's vision of a nonsequential way of storing data. Now republished by Microsoft Press, COMPUTER LIB/DREAM MACHINES brings this wildly Utopian introduction to computers to a new generation. Includes new commentaries, insights, and reconsiderations.

336 pages, softcover *$18.95*
Order Code: 86-96031

Microsoft Press books and Tempus books are available wherever fine books are sold. Credit card orders can be placed by calling 1-800-638-3030 (in Maryland call collect 824-7300).

Tempus Books is an imprint of Microsoft Press.

The manuscript for this book was prepared and submitted to Microsoft Press in electronic form. Text files were processed and formatted using Microsoft Excel for the Apple Macintosh.

Cover design by Greg Hickman
Color separations by Wescan Color Corporation
Principal table formatting by Mark Dodge

All table composition by Microsoft Press in Helvetica, using Microsoft Excel for the Macintosh and output by the Linotronic 300 laser imagesetter.

THE PROGRAMMER'S APPLE® MAC® SOURCEBOOK

Thom Hogan

At last! An information-packed book that ends the repeated frustration of scouring your reference library for key pieces of technical data. This unique resource integrates facts about the Macintosh that used to be scattered in dozens of sources into one convenient reference. With more than 300 pages of charts and tables, all thoroughly indexed, THE PROGRAMMER'S APPLE MAC SOURCEBOOK is your primary reference to Macintosh® Plus, SE, and II hardware and software, with specific details:

Data formats ▪ The many software and disk resource formats ▪ Interface toolbox managers, including Color Quick Draw™, Window Manager, Font Manager, and Menu Manager ▪ NuBus™ Slot Manager ▪ Operating system calls ▪ Machine summaries of the keyboards, memory organization, CPUs, registers, protocols, pinouts and connectors, and printers ▪ Hypercard®, Installer, System, and Finder™ ▪ Video driver information ▪ Hexadecimal and binary conversions ▪ ASCII codes ▪ And more

Included with each table are data sources, cross-references to related information in the book, and notes to help you use the information.

THE PROGRAMMER'S APPLE MAC SOURCEBOOK. The first place to turn for immediate, accurate information about your Macintosh computer, its software, and its hardware.

U. S. A.	**$22.95**
U. K.	£21.95
Austral.	$34.95
(recommended)	

ISBN 1-55615-168-3

9 781556 151682 5229