

# Index

---

## Numerals

---

32-bit clean 1-4  
680x0 applications 1-6 to 1-12  
    porting to PowerPC 1-15 to 1-19, 1-31 to 1-34, 1-57 to 1-65, 1-68 to 1-72, 2-21 to 2-26, 3-12 to 3-13, 4-6 to 4-9  
    structure of 1-32  
680x0 compatibility issues. *See* 68LC040 Emulator  
680x0 context block 1-8, 1-59  
680x0 registers  
    *.See also* A0 register, A5 register, A6 register, A7 register  
    unsupported results 1-10  
680x0 run-time environment 1-57 to 1-59  
    data alignment 1-63 to 1-65  
68851 paged memory-management unit 1-9  
68881 floating-point unit 1-9  
68882 floating-point unit 1-9  
    68LC040 Emulator 1-3, 1-6 to 1-12. *See also* Mixed Mode Manager  
    address error exceptions 1-10  
    bus error exceptions 1-11  
    byte smearing 1-12  
    dynamic bus sizing 1-12  
    floating-point instructions 1-9  
    instruction cache 1-10  
    instruction timings 1-9  
    NOP instruction 1-12  
    PMMU 1-9  
    reserved fields 1-10  
    unavailable instructions 1-9  
    undefined results 1-10  
    virtual memory 1-9

---

## A

---

A0 register, and the Vertical Retrace Manager 1-62  
A5 register, setting and restoring 1-60 to 1-63  
A5 world 1-57 to 1-63  
    and table of contents 1-28  
A6 register 1-42  
A7 register 1-42  
accelerated resources 1-23, 1-34 to 1-40  
    calling at interrupt time 2-26  
    data section in 1-38  
    limitations on 1-38 to 1-40

    and main symbols 1-38  
    and termination routines 1-38  
    using global data in 1-39 to 1-40  
action procedures. *See* control action procedures  
address error exceptions, emulator compatibility issues 1-10  
alert boxes, displayed by PowerPC applications 1-34  
alias resources 3-31  
alignment. *See* data alignment  
A-line instructions 1-8  
    'alis' resource type 3-31  
ANSI-compliant source code ix, 1-65  
APDA xv  
AppleShare servers 1-55  
    'APPL' file type 1-21  
application extensions. *See also* fragments  
    defined 1-21  
    application global variables 1-58  
    application parameters 1-58  
    application partitions  
        automatic resizing of 1-55  
    applications 1-21. *See also* fragments  
        file type 1-21  
        length of fragment 3-31  
        location of fragment 3-31  
        specifying instruction set architecture 3-30  
        specifying library directory 3-31  
        specifying stack size 1-60, 3-31  
AppLimit global variable 1-60  
A-traps. *See* A-line instructions

---

## B

---

backing-store file 1-53  
backing volume. *See* paging devices  
BCLR instruction 1-12  
binding 1-25, 1-28  
bit numbering conventions xiii  
block headers 1-69  
BlockMove procedure 1-70  
BSET instruction 1-12  
BuildFatRoutineDescriptor macro 2-24  
BuildRoutineDescriptor macro 2-23 to 2-24  
bus error exceptions, emulator compatibility issues 1-11  
bus sizing. *See* dynamic bus sizing  
byte smearing, emulator compatibility issues 1-12

## C

---

CAAR. *See* Cache Address Register

cache

- Emulator compatibility issues 1-10

Cache Address Register (CAAR), Emulator

- compatibility issues 1-10

Cache Control Register (CACR), Emulator

- compatibility issues 1-10

CACR. *See* Cache Control Register

calling conventions 1-41 to 1-47

- 680x0 1-42 to 1-43
- C routines 1-43, 2-30
- Operating System routines 1-43
- Pascal routines 1-43, 2-30
- PowerPC 1-43 to 1-47
- register-based routines 2-30
- .See also* procedure information
- selector-based C routines 2-30
- selector-based Pascal routines 2-30
- special cases. *See* special case routines
- specifying 2-30 to 2-32
- THINK C routines 2-30

CALLM instruction 1-9

CalloSTrapUniversalProc function 1-67, 2-42 to 2-43

CallUniversalProc function 1-37, 1-67, 2-42

CCR. *See* Condition Code Register

'CDEF' resources 1-36

'cfrg' resource type 1-31 to 1-34, 3-12 to 3-13, 3-28 to 3-31

CloseConnection function 1-41, 3-23 to 3-24

closing resource forks 1-70

code, self-modifying 1-53

code fragment information records 3-29 to 3-31

Code Fragment Loader 1-22

Code Fragment Manager 1-22, 3-3 to 3-35

- data structures 3-15 to 3-18
- reading code fragment resources 3-13
- resources 3-28 to 3-31
- routines 3-18 to 3-26

code fragment resources 1-31 to 1-34, 3-12 to 3-13, 3-28 to 3-31

code fragments. *See* fragments

code patches. *See* patches

code resources. *See* executable resources

code sections 1-23

code types. *See* instruction set architectures

compact discs 1-55

compatibility issues. *See* 68LC040 Emulator

compile-time libraries. *See* definition libraries

completion routines 1-18

Condition Code Register (CCR)

- during mode switches 2-14
- specifying in procedure information 2-20

Condition Register (CR) 1-45, 1-46

connection IDs 3-5

connections 3-5

containers

- defined 1-21, 3-4
- specifying location of 3-31

context blocks. *See* 680x0 context blocks

contexts 1-51

control action procedures 1-16 to 1-18

control definition functions 1-16, 1-36

control panels

- Memory 1-68

cooperative multitasking environment 1-4

coprocessors 1-9

counting symbols 3-14, 3-25 to 3-26

CountSymbols function 3-14, 3-25 to 3-26

cross-mode call

- See* explicit cross-mode call, implicit cross-mode call

CR. *See* Condition Register

## D

---

data, exchanging between PowerPC and 680x0

- environments 1-64 to 1-65

data, global. *See* global data

data alignment 1-63 to 1-65

data forks 1-21, 1-30, 1-31 to 1-34

data instantiation

- global 1-51
- per-context 1-51
- per-load 1-52

data sections

- and accelerated resources 1-38
- defined 1-23

Debugger routine

- calling within an exception handler 4-9

DebugStr routine

- calling within an exception handler 4-9

default stack size 1-60, 3-31

definition procedures. *See* control definition functions, list definition procedures, menu definition procedures, window definition functions

definition versions 3-8, 3-30

detaching resources 1-70

DetachResource procedure 1-70

device drivers

- and the 68LC040 Emulator 1-11 to 1-12

DiskFragment data type 3-17

disk location records 3-17 to 3-18

DISPATCHED\_STACK\_ROUTINE\_PARAMETER

- macro 2-50

DISPATCHED\_STACK\_ROUTINE\_SELECTOR\_SIZE

- macro 2-50

DisposeHandle procedure 1-69  
 DisposePtr procedure 1-70  
 DisposeRoutineDescriptor function 1-19, 2-21, 2-41  
 disposing of memory blocks 1-69  
 disposing of pictures 1-69  
 draw hook routines, specifying calling conventions of 2-32  
 drop-ins. *See* application extensions  
 dynamically linked libraries. *See* import libraries  
 dynamic bus sizing  
     Emulator compatibility issues 1-12

## E

---

emulator. *See* 68LC040 Emulator  
 epilog code 1-46  
 event filter functions 1-18  
 exception codes. *See* exceptions, types of  
 exception contexts 4-4  
 exception frames  
     created by 68LC040 Emulator 1-11  
 exception handlers  
     and the Red Zone 1-47  
     defined 4-3  
     installing 1-57, 4-6 to 4-7  
     limitations on 4-9  
     removing 4-7  
     writing 4-7 to 4-9  
 ExceptionInformation data type 4-7, 4-16  
 exception information records 4-7, 4-16  
 Exception Manager 1-47, 4-3 to 4-22  
     application-defined routines in 4-17 to 4-18  
     constants in 4-9 to 4-11  
     data structures in 4-12 to 4-16  
     routines in 4-17  
 exceptions  
     680x0 bus error 1-11  
     defined 4-3  
     types of 4-5 to 4-6, 4-9 to 4-11  
 exchanging data between PowerPC and 680x0 environments 1-64 to 1-65  
 executable resources 1-34 to 1-41  
     *See also* accelerated resources, private resources  
 ExitToShell procedure 2-41  
 explicit cross-mode calls 2-8  
 exported symbols. *See* exports  
 exports 1-23, 3-4  
     getting information about 3-14  
 Extended Common Object File Format (XCOFF) 1-22, 1-30  
 Extensions folder 3-6, 3-7

extensions. *See* application extensions and system extensions  
 external code 2-4 to 2-5

## F

---

fake definition resources. *See* stub definition resources  
 fake handles 1-70  
 fake pointers 1-70  
 fat applications 1-33 to 1-34  
 fat patches 1-66 to 1-68, 1-71  
 fat resources 1-38, 1-71, 2-25  
 fat routine descriptors 2-24, 2-25  
 file and directory registry 3-6 to 3-7  
 file forks. *See* data forks, resource forks  
 file mapping 1-53 to 1-55  
 file types  
     'APPL' 1-21  
     'sh1b' 1-21, 3-6, 3-10  
 finding symbols 1-38, 3-14, 3-24 to 3-26  
 FindSymbol function 1-38, 1-41, 3-24 to 3-25  
 F-line instructions 1-8  
 floating-point data types 1-65  
 floating-point exceptions  
     handling 4-3  
 floating-point information records 4-14  
 floating-point instructions  
     Emulator compatibility issues 1-9  
 floating-point parameters 1-72  
 floating-point registers 1-43, 1-47, 1-47 to 1-50, 1-72, 4-4, 4-15  
 Floating-Point Status and Control Register (FPSCR) 4-14 to 4-15  
 floppy disks 1-55  
 flushing caches 1-10, 1-70  
 forks. *See* data forks, resource forks  
 FPSCR. *See* Floating-Point Status and Control Register  
 FP. *See* frame pointer  
 FPUInformation data type 4-14  
 fragment initialization blocks 3-15 to 3-16  
 fragment location records 3-16 to 3-17  
 FragmentLocator data type 3-16  
 fragments 1-20 to 1-41, 3-4 to 3-5  
     defined 1-5, 1-21, 3-4  
     finding symbols in 3-24 to 3-26  
     kinds of 1-21  
     loading 3-10 to 3-12, 3-19 to 3-22  
     special routines in 1-29 to 1-30, 3-26 to 3-28  
     specifying names of 3-31  
     specifying size of 3-31  
     storing 1-30 to 1-34  
     structure of 1-22 to 1-23  
     unloading 3-23 to 3-24

frame pointer 1-42  
 frames. *See* stack frames or switch frames  
 free blocks 1-70  
 function prototypes 1-72, 2-30

## G

---

general-purpose registers 1-8, 1-26, 1-41, 1-43, 1-45,  
 1-47, 1-47 to 1-50, 1-72, 4-4, 4-8, 4-12 to 4-14  
 Gestalt function 1-25, 1-57  
 Get1Resource function 3-21  
 GetApplLimit function 1-60, 1-70, 3-31  
 GetCurrentISA function 2-44  
 GetDiskFragment function 3-11, 3-19 to 3-21  
 GetIndSymbol function 3-14, 3-26  
 GetMemFragment function 3-11, 3-21 to 3-22  
 GetNextEvent filter procedures, specifying calling  
 conventions of 2-32  
 GetPicture function 1-69  
 GetSharedLibrary function 3-10, 3-22 to 3-23  
 global data  
     in accelerated resources 1-39 to 1-40  
 global instantiation 1-51  
 global variables. *See* application global variables,  
     QuickDraw global variables, and system global  
     variables  
 grow-zone functions 1-18  
     specifying procedure information for 2-17 to 2-18

## H

---

handles, fake 1-70  
 header files. *See* universal interface files  
 head patches 1-68  
 hit test hook routines, specifying calling conventions  
     of 2-32  
 hybrid environment. *See* mixed environment  
 HyperCard extensions 1-36

## I

---

implementation versions 3-8, 3-30  
 implicit cross-mode calls 2-8  
 imported symbols. *See* imports  
 import libraries 1-50 to 1-52  
     advantages of 1-51  
     checking versions 3-7 to 3-10  
     data instantiation 1-51 to 1-52  
     defined 1-21

definition version 3-8  
 file and directory registry 3-6 to 3-7  
 file type 1-21, 3-6, 3-10  
 implementation version 3-8  
 length of fragment 3-31  
 load directories 3-7  
 location of fragment 3-31  
 ROM registry 3-6  
 search order 3-5 to 3-7  
     *See also* fragments  
     specifying definition version 3-30  
     specifying implementation version 3-30  
     specifying instruction set architecture 3-30  
     specifying update levels 3-30  
 imports 1-21, 3-4  
 imports. *See also* soft imports  
 InitBlock data type 3-15  
 InitGraf procedure 1-59  
 initialization blocks. *See* fragment initialization blocks  
 initialization routines 3-15 to 3-18, 3-27  
     defined 1-30  
 in-place data instantiation 1-38  
 input/output, accessing memory-mapped  
     locations 1-11 to 1-12  
*Inside Macintosh*  
     bit numbering conventions xii to xiii  
     chapter format xi  
     format conventions xii  
     format of parameter blocks xiv  
 InstallExceptionHandler function 4-17  
 instantiation. *See* global instantiation, per-context  
     instantiation, and per-load instantiation  
 instruction cache 1-10, 1-70  
 instruction set architectures  
     constants for 2-35 to 2-36  
     defined 1-13  
     determining 2-44  
     specifying for an application 3-30  
     specifying for an import library 3-30  
 instruction timings 1-9  
 interface files. *See* universal interface files  
 interrupts. *See* exceptions  
 interrupt time  
     calling accelerated resources 2-26  
     calling Memory Manager 1-70  
 I/O. *See* input/output

## J

---

jump tables 1-58

## K

---

KillPicture procedure 1-69

## L

---

'LDEF' resources 1-36  
 leaf procedures 1-46  
 libraries. *See* import libraries  
 library directories 3-6, 3-31  
 line-start recalculation routines, specifying calling conventions of 2-32  
 linkage area 1-44  
 Link Register 2-11  
 list definition procedures 1-35 to 1-36  
 LMGetCurDirStore function 1-57  
 load directories 3-7  
 loading code fragments 3-10 to 3-12, 3-19 to 3-23  
 location records. *See* fragment location records  
 low-memory global variables. *See* system global variables  
 LR. *See* Link Register

## M

---

MachineInformation data type 4-7, 4-12  
 machine information records 4-7, 4-12  
 Macintosh Programmer's Workshop xiv, 1-32, 1-38, 1-57, 1-65, 2-26, 2-30  
 main routines 3-27  
     and accelerated resources 1-38  
     defined 1-30  
 main symbols 3-19, 3-21, 3-22  
     and accelerated resources 1-38  
     defined 1-30  
 MakePEF tool 1-26, 1-38  
 'MDEF' resources 1-36  
 MemFragment data type 3-17  
 memory  
     organization of 1-52 to 1-65  
 memory blocks  
     disposing of 1-69  
 Memory control panel 1-68  
 MemoryExceptionInformation data type 4-15  
 memory exception records 4-15  
 memory location records 3-17  
 Memory Manager 1-5, 1-68 to 1-70  
     at interrupt time 1-70  
     disposing of blocks 1-69  
     private data structures 1-69  
 memory operations

types of 4-11  
 memory reference codes 4-11  
 menu bar hook routines, specifying calling conventions of 2-32  
 menu definition procedures 1-36  
 mini-A5 world 1-60  
 mixed environment 1-3, 1-4  
 Mixed Mode Manager 1-4, 1-13 to 1-19, 2-3 to 2-50  
     constants in 2-27 to 2-36  
     data structures in 2-36 to 2-38  
     defined 1-13, 2-3  
     introduced 2-4  
     limitations of 2-21  
     routines in 2-38 to 2-44  
     *See also* 68LC040 Emulator  
     *See also* mixed environment, routine descriptors  
     *See also* mode switches  
 mode switches 2-7 to 2-14  
     defined 1-13  
     in patches 1-66  
     overhead 1-66  
 MOVE instruction 1-12  
 MPW. *See* Macintosh Programmer's Workshop

## N

---

nanokernel 1-4  
 NewControlActionProc function 1-18  
 NewFatRoutineDescriptor function 2-21, 2-40 to 2-41  
 NewPtr function 1-67  
 NewRoutineDescriptor function 2-15, 2-21, 2-39 to 2-40  
 NOP instruction  
     Emulator compatibility issues 1-12  
 NSetTrapAddress procedure 1-67  
 null events 1-71 to 1-72

## O

---

opcodes. *See* operation codes  
 operation codes 1-8

## P

---

paged memory management unit  
     Emulator compatibility issues 1-9  
 paging devices 1-55  
 parameter area 1-44

- parameter blocks
  - format of `xiv`
- parameter lists, variable 1-73
- parameter passing 1-47 to 1-50
- patches 1-18, 1-66 to 1-68
  - fat 1-66 to 1-68
  - head 1-68
  - tail 1-68
- patching
  - selector-based traps 1-68
- PC. *See* program counter
- PEF. *See* Preferred Executable Format
- per-context instantiation 1-51
- performance 1-70 to 1-73
  - avoiding mode switches 1-71 to 1-72
  - passing parameters 1-72 to 1-73
  - using fat resources 1-71
- per-load instantiation 1-52
- pictures
  - disposing of 1-69
- PMMU. *See* paged memory-managment unit
- pointer-based function calls 1-29
- pointers, fake 1-70
- porting 680x0 applications to PowerPC. *See* 680x0 applications, porting to PowerPC
- PowerPC applications
  - structure of 1-31 to 1-32
- PowerPC microprocessor ix, 1-4
  - floating-point registers 1-43, 1-47 to 1-50, 1-72, 4-4, 4-15
  - general-purpose registers 1-8, 1-26, 1-41, 1-43, 1-45, 1-47, 1-47 to 1-50, 1-72, 4-4, 4-8, 4-12 to 4-14
  - special-purpose registers 1-41, 1-44 to 1-46, 4-4, 4-8, 4-12
- PowerPC run-time environment 1-19 to 1-65
  - application partitions 1-57 to 1-63
  - data alignment 1-63 to 1-65
  - organization of memory in 1-52 to 1-65
  - system partition 1-56 to 1-57
- PowerPC. *See* PowerPC microprocessor
- pragma statements 1-64
- Preferred Executable Format (PEF) 1-22, 1-30
- prepare 1-22
- private resources 1-36, 1-40 to 1-41
- procedure information
  - constants for 2-27 to 2-33
  - defined 1-16, 2-15
  - number of specifiable parameters 2-17, 2-20
  - specifying 2-14 to 2-21
- procedure pointers 2-5 to 2-7
- Process Manager
  - reading code fragment resources 3-12
- `ProcInfoType`. *See* procedure information
- `ProcPtr`. *See* procedure pointer
- program counter 1-8, 1-11, 4-8, 4-12

- prolog code 1-45
- protocol handlers, specifying calling conventions
  - of 2-32
- prototypes. *See* function prototypes

## Q

---

- `QDGlobals` data type 1-59
- QuickDraw global variables 1-58 to 1-60

## R

---

- Red Zone 1-46 to 1-47
- reentrancy
  - in exception handlers 4-9
- `REGISTER_RESULT_LOCATION` macro 2-18, 2-50
- `REGISTER_ROUTINE_PARAMETER` macro 2-18, 2-50
- `RegisterInformation` data type 4-8, 4-12 to 4-14
- register information records 4-12 to 4-14
- registers. *See* PowerPC microprocessor, 680x0 registers
- `ReleaseResource` procedure 1-69
- resource-based code
  - executing 2-24 to 2-26
  - See also* fat resources
- resource forks 1-31 to 1-34
  - closing 1-70
- resources
  - accelerated. *See* accelerated resources
  - detaching 1-70
  - fat 1-71
  - private. *See* private resources
  - stub. *See* stub definition resources
- resource types
  - '`alis`' 3-31
  - '`CDEF`' 1-36
  - '`cfrg`' 1-31 to 1-34, 3-12 to 3-13, 3-28 to 3-31
  - '`LDEF`' 1-36
  - '`MDEF`' 1-36
  - '`WDEF`' 1-36
  - '`XCMD`' 1-36
- `RESULT_SIZE` macro 1-16, 2-16, 2-50
- `Rez` 1-32, 1-38, 2-26, 3-12, 3-13, 3-28, 3-30, 3-31
- ROM registry 3-6
- `RoutineDescriptor` data type 2-37 to 2-38
- routine descriptor flags 2-27
- routine descriptors 1-15 to 1-19, 2-6 to 2-7, 2-37 to 2-38
  - creating 2-39 to 2-41
  - defined 1-15, 2-6
  - disposing of 1-19, 2-41
  - executing code with 2-42 to 2-43
  - fat 2-24, 2-25

- global 2-21
- local 2-21 to 2-22
- See also* universal procedure pointers
- static 2-22 to 2-24
- RoutineRecord data type 2-36
- routine records 1-15 to 1-16, 2-36 to 2-37
- RTE instruction 1-11
- RTM instruction 1-9
- ROTC. *See* Table of Contents Register
- run-time environment
  - defined 1-20
- run-time environment. *See* PowerPC run-time environment and 680x0 run-time environment
- run-time libraries. *See* implementation libraries

## S

---

- SANE. *See* Standard Apple Numerics Environment
- saved registers area 1-45
- sections 1-22
  - See also* code sections and data sections
- SegmentedFragment data type 3-18
- segment location records 3-18
- Segment Manager 1-32
- selector-based traps 1-68
- self-modifying code 1-53
- SetA5 function 1-62 to 1-63
- SetApplLimit procedure 1-60, 1-69, 1-70, 3-31
- SetCurrentA5 function 1-63
- SetGrowZone procedure 1-69
- SetOSTrapAddress procedure 1-67
- SetToolTrapAddress procedure 1-67
- SetTrapAddress procedure 1-67
- shared libraries. *See* import libraries
- 'shlb' file type 1-21, 3-6, 3-10
- 68881 floating-point unit 1-9
- 68882 floating-point unit 1-9
- 68851 paged memory-management unit 1-9
- 680x0 registers
  - .See also* A0 register, A5 register, A6 register, A7 register
  - unsupported results 1-10
- SIZE\_CODE macro 1-16, 2-50
- smearing. *See* byte smearing
- socket listeners, specifying calling conventions of 2-32
- soft imports 1-25 to 1-26
- SPECIAL\_CASE\_PROCINFO macro 2-50
- special case routines 2-30 to 2-32
- special-purpose registers 1-41, 1-44 to 1-46, 4-4, 4-8, 4-12
- Special Status Word (SSW) 1-11
- split traps 1-68
- SP. *See* stack pointer

- SSW. *See* Special Status Word
- stack, specifying minimum size of 1-60, 3-31
- STACK\_ROUTINE\_PARAMETER macro 1-16, 2-50
- stack frames 1-41, 1-42 to 1-47
  - parameter area 1-44
  - See* switch frames
- stack pointer 1-8, 1-42, 2-10
- stale instructions 1-10
- Standard Apple Numerics Environment (SANE) 1-9
- stub definition resources 1-35
- switches. *See* mode switches
- switch frames
  - 680x0-to-PowerPC 2-10 to 2-12
  - PowerPC-to-680x0 2-13 to 2-14
- symbols 3-4
  - counting 3-14, 3-25 to 3-26
  - finding 1-38, 3-14, 3-24 to 3-26
- System 7.1 1-4
- system extensions
  - defined 1-21
- system global variables 1-56 to 1-57, 1-69
- system partition 1-56 to 1-57
- system software
  - for PowerPC processor-based Macintosh computers 1-4 to 1-6
  - patching 1-66 to 1-68

## T

---

- table of contents 1-26 to 1-29
  - defined 1-26
  - maximum size of 1-29
- Table of Contents Register (ROTC) 1-26, 1-27, 1-29, 1-45, 1-46, 2-11
- tail patches 1-68
- temporary memory 1-55
- termination routines 3-28
  - and accelerated resources 1-38
  - defined 1-30
- text display routines, specifying calling conventions of 2-32
- text width hook routines, specifying calling conventions of 2-31
- THINK C calling conventions 2-30
- 32-bit clean 1-4
- Time Manager tasks 1-18, 1-60
- TOC. *See* table of contents
- tools. *See* application extensions
- TrackControl procedure 1-17, 2-21
- transition vectors 1-26 to 1-27
  - and exception handlers 4-17
  - defined 1-26, 2-5
- trap patches. *See* patches

traps  
   selector-based 1-68  
   split 1-68

## U

---

universal interface files 1-18 to 1-19, 1-57, 1-65, 2-6 to 2-7, 2-15, 2-17  
 universal procedure pointers 1-17 to 1-19, 2-6 to 2-7, 2-37  
   and accelerated resources 1-37, 2-24 to 2-26  
   and fat patches 1-66  
   and universal interface files 2-15  
   defined 2-6  
   executing code with 2-42 to 2-43  
   *.See also* routine descriptors  
   used in stub definition functions 1-36  
   using 2-21 to 2-22  
 unloading code fragments 3-23 to 3-24  
 UnloadSeg procedure 1-6  
 update levels  
   specifying for an import library 3-30  
 USESRROUTINEDESCRIPTORS compiler variable 2-14, 2-39

## V

---

variable parameter lists 1-73  
 VBL tasks 1-18, 1-60 to 1-63  
 vectors. *See* transition vectors  
 versions  
   of import libraries 3-7 to 3-10  
   of routine descriptor 2-38  
 Vertical Retrace Manager 1-61 to 1-63  
 virtual memory 1-53 to 1-55  
   Emulator support for 1-9  
 Virtual Memory Manager 1-4, 1-53

## W

---

WaitNextEvent function 1-71  
 'WDEF' resources 1-36  
 weak imports. *See* soft imports  
 width hook routines, specifying calling conventions  
   of 2-31  
 window definition functions 1-36  
 word sizes xiii, 1-63

## X

---

'XCMD' resources 1-36  
 XCOFF. *See* Extended Common Object File Format

## Z

---

zone headers 1-69



