```
; QuickDraw Equates -- This file defines the public equates for QuickDraw.

; Transfer modes

srcCopy         EQU   0
srcOr           EQU   1
srcXor          EQU   2
srcBic          EQU   3
notSrcCopy      EQU   4
notSrcOr        EQU   5
notSrcXor       EQU   6
notSrcBic       EQU   7
patCopy         EQU   8
patOr           EQU   9
patXor          EQU   10
patBic          EQU   11
notPatCopy      EQU   12
notPatOr        EQU   13
notPatXor       EQU   14
notPatBic       EQU   15

; Definitions for Font Style Bits (right to left)

boldBit         EQU   0
italicBit       EQU   1
ulineBit        EQU   2
outlineBit      EQU   3
shadowBit       EQU   4
condenseBit     EQU   5
extendBit       EQU   6


; FontInfo      record

ascent          EQU   0         ; ascent [word]
descent         EQU   2         ; descent [word]
widMax          EQU   4         ; maximum width [word]
leading         EQU   6         ; leading [word]


; Point structure

v               EQU   0         ; vertical coordinate [word]
h               EQU   2         ; horizontal coordinate [word]

; Rectangle structure

topLeft         EQU   0         ; upper left corner [point]
botRight        EQU   4         ; lower right corner [point]
top             EQU   0         ; top coordinate [word]
left            EQU   2         ; left coordinate [word]
```

```
bottom          EQU   4       ; bottom coordinate [word]
right           EQU   6       ; right coordinate [word]

; Bitmap Structure

baseAddr        EQU   0       ; bitmap base address [pointer]
rowBytes        EQU   4       ; row bytes (must be even) [word]
```

```
bounds          EQU   6          ; bounding box [rectangle]
bitmapRec       EQU   14         ; size of a bitmap

; Cursor Structure

data            EQU   0          ; visible bits [32 bytes]
mask            EQU   $20        ; mask bits [32 bytes]
hotSpot         EQU   $40        ; relative origin [point]
cursRec         EQU   68         ; size of a cursor

; PenState record

psLoc           EQU   0          ; pen location [point]
psSize          EQU   4          ; pen size [point]
psMode          EQU   8          ; pen mode [word]
psPat           EQU   10         ; pen [pattern]
psRec           EQU   18         ; size of pen state

; Polygon record

polySize        EQU   0          ; total bytes [word]
polyBBox        EQU   2          ; bounding box [rectangle]
polyPoints      EQU   10         ; vertices [Points]

; Region Structure

rgnSize         EQU   0          ; total bytes [word]
rgnBBox         EQU   2          ; bounding box [rectangle]
rgnData         EQU   10         ; region data [array]

; Picture Structure

picSize         EQU   0          ; total bytes [word]
picFrame        EQU   2          ; bounding box [rectangle]
picData         EQU   10         ; picture byte codes [array]

; QDProcs structure

textProc        EQU   0          ; [pointer]
lineProc        EQU   4          ; [pointer]
rectProc        EQU   8          ; [pointer]
rRectProc       EQU   $C         ; [pointer]
ovalProc        EQU   $10        ; [pointer]
arcProc         EQU   $14        ; [pointer]
polyProc        EQU   $18        ; [pointer]
rgnProc         EQU   $1C        ; [pointer]
bitsProc        EQU   $20        ; [pointer]
commentProc     EQU   $24        ; [pointer]
txMeasProc      EQU   $28        ; [pointer]
getPicProc      EQU   $2C        ; [pointer]
putPicProc      EQU   $30        ; [pointer]
```

```
qdProcsRec          EQU   $34         ; size of QDProcs record

; GrafPort Structure

device              EQU   $0          ; device code [word]
portBits            EQU   $2          ; port's bitmap [bitmap]
portBounds          EQU   $8          ; bounding box of bitmap [rect]
```

```
portRect        EQU   $10         ; port's rectangle [rect]
visRgn          EQU   $18         ; visible region [handle]
clipRgn         EQU   $1C         ; clipping region [handle]
bkPat           EQU   $20         ; background [pattern]
fillPat         EQU   $28         ; fill [pattern]
pnLoc           EQU   $30         ; pen location [point]
pnSize          EQU   $34         ; pen size [point]
pnMode          EQU   $38         ; pen mode [word]
pnPat           EQU   $3A         ; pen [pattern]
pnVis           EQU   $42         ; pen visible [word]
txFont          EQU   $44         ; text font [word]
txFace          EQU   $46         ; text face [word]
txMode          EQU   $48         ; text mode [word]
txSize          EQU   $4A         ; text size [word]
spExtra         EQU   $4C         ; space extra [fixed]
fgColor         EQU   $50         ; foreground color mask [long]
bkColor         EQU   $54         ; background color mask [long]
colrBit         EQU   $58         ; color bit [word]
patStretch      EQU   $5A         ; pattern stretch [word]
picSave         EQU   $5C         ; picture being saved [handle]
rgnSave         EQU   $60         ; region being saved [handle]
polySave        EQU   $64         ; polygon being saved [handle]
grafProcs       EQU   $68         ; QDProcs array [pointer]
portRec         EQU   $6C         ; size of grafport

; QuickDraw Global Variables

GrafGlobals     EQU   0            ; A5 offset to globptr

thePort         EQU   0           ;GrafPtr
white           EQU   thePort-8   ;Pattern
black           EQU   white-8     ;Pattern
gray            EQU   black-8     ;Pattern
ltGray          EQU   gray-8      ;Pattern
dkGray          EQU   ltGray-8    ;Pattern
arrow           EQU   dkGray-68   ;Cursor
screenBits      EQU   arrow-14    ;BitMap
randSeed        EQU   screenBits-4 ;LONGINT

; Color Separation

normalBit       EQU   0           ; normal screen mapping
inverseBit      EQU   1           ; inverse screen mapping
redBit          EQU   4           ; RGB additive mapping
greenBit        EQU   3           ;  for photos from screen
blueBit         EQU   2
cyanBit         EQU   8           ; CMYBk subtractive mapping
magentaBit      EQU   7           ;  for ink jet printer
```

```
yellowBit       EQU  6
blackBit        EQU  5
blackColor      EQU  33
whiteColor      EQU  30
redColor        EQU  205
greenColor      EQU  341
blueColor       EQU  409
```

```
cyanColor               EQU   273
magentaColor            EQU   137
yellowColor             EQU   69


; Standard Picture Comments

picLParen               EQU   0
picRParen               EQU   1

; QuickDraw verbs

frame                   EQU   0
paint                   EQU   1
erase                   EQU   2
invert                  EQU   3
fill                    EQU   4

; QuickDraw private global variables

wideOpen                EQU   randSeed-4        ;RgnHandle
wideMaster              EQU   wideOpen-4        ;RgnPtr
wideData                EQU   wideMaster-10     ;Fake Region
rgnBuf                  EQU   wideData-4        ;PointsHandle
rgnIndex                EQU   rgnBuf-2          ;INTEGER
rgnMax                  EQU   rgnIndex-2        ;INTEGER
playPic                 EQU   rgnMax-4          ;Long
qdSpare0                EQU   playPic-2         ;unused word
thePoly                 EQU   qdSpare0-4        ;POLYHANDLE
polyMax                 EQU   thePoly-2         ;INTEGER
patAlign                EQU   polyMax-4         ;Point
fontAdj                 EQU   patAlign-4        ;Fixed Point
fontPtr                 EQU   fontAdj-4         ;long, ^FMOutput record
playIndex               EQU   fontPtr-4         ;long
fontData                EQU   playIndex-22      ;unused word
lastGrafGlob            EQU   fontData
grafSize                EQU   4-lastGrafGlob    ;total size in bytes


;*********FROM HERE TO THE ENDIF IS ALL THE NEW STUFF FOR COLOR QUICKDRAW

hiliteBit               EQU   7         ; flag bit in HiliteMode (lowMem flag)

;                       Equates for resource ID's

defQDColors             EQU   127       ; resource ID of clut for default QDColors


;
;                       PixMap field offsets
;
```

```
pmBaseAddr      EQU   $0      ; [long]
pmNewFlag       EQU   $4      ; [1 bit]   upper bit of rowbytes is flag
pmRowBytes      EQU   $4      ; [word]
pmBounds        EQU   $6      ; [rect]
pmVersion       EQU   $E      ; [word]    pixMap version number
pmPackType      EQU   $10     ; [word]    defines packing format
```

```
pmPackSize      EQU   $12       ; [long]    size of pixel data
pmHRes          EQU   $16       ; [fixed]   h. resolution (ppi)
pmVRes          EQU   $1A       ; [fixed]   v. resolution (ppi)
pmPixelType     EQU   $1E       ; [word]    defines pixel type
pmPixelSize     EQU   $20       ; [word]    # bits in pixel
pmCmpCount      EQU   $22       ; [word]    # components in pixel
pmCmpSize       EQU   $24       ; [word]    # bits per field
pmPlaneBytes    EQU   $26       ; [long]    offset to next plane
pmTable         EQU   $2A       ; [long]    color map
pmReserved      EQU   $2E       ; [long]    MUST BE 0

pmRec           EQU   $32       ; size of pixMap record

;               PixPat field offsets
;

patType         EQU   $0        ; [word] type of pattern
patMap          EQU   $2        ; [long] handle to pixmap
patData         EQU   $6        ; [long] handle to data
patXData        EQU   $A        ; [long] handle to expanded pattern data
patXValid       EQU   $E        ; [word] flags whether expanded pattern valid
patXMap         EQU   $10       ; [long] handle to expanded pattern data
pat1Data        EQU   $14       ; [8 bytes] old-style pattern/RGB color

ppRec           EQU   $1C       ; size of pixPat record


;               Pattern types
;

oldPat          EQU   0         ; foreground/background pattern
newPat          EQU   1         ; self-contained color pattern
ditherPat       EQU   2         ; rgb value to be dithered

oldCrsrPat      EQU   $8000     ; old-style cursor
cCrsrPat        EQU   $8001     ; new-style cursor


;               additional offsets in a color GrafPort
;

portPixMap      EQU   portBits      ; [long] pixelMap handle
portVersion     EQU   portPixMap+4  ; [word] port version number
grafVars        EQU   portVersion+2 ; [long] handle to more fields
chExtra         EQU   grafVars+4    ; [word] character extra
pnLocHFrac      EQU   chExtra+2     ; [word] pen fraction

bkPixPat        EQU   bkPat         ; [long] handle to bk pattern
rgbFgColor      EQU   bkPixPat+4    ; [6 bytes] RGB components of fg color
rgbBkColor      EQU   rgbFgColor+6  ; [6 bytes] RGB components of bk color
```

```
pnPixPat        EQU   $3A          ; [long] handle to pen's pattern
fillPixPat      EQU   pnPixPat+4    ; [long] handle to fill pattern


;               GDevice field offsets
;
```

```
gdRefNum        EQU   $0        ; [word] unitNum of driver
gdID            EQU   $2        ; [word] client ID for search procs
gdType          EQU   $4        ; [word] fixed/CLUT/direct
gdITable        EQU   $6        ; [long] handle to inverse table
gdResPref       EQU   $A        ; [word] preferred resolution for inverse tables
gdSearchProc    EQU   $C        ; [long] search proc (list?) pointer
gdCompProc      EQU   $10       ; [long] complement proc (list?) pointer
gdFlags         EQU   $14       ; [word] grafDevice flags word
gdPMap          EQU   $16       ; [long] handle to pixMap describing device
gdRefCon        EQU   $1A       ; [long] reference value
gdNextGD        EQU   $1E       ; [long] handle of next gDevice
gdRect          EQU   $22       ; [rect] device's bounds in global coordinates
gdMode          EQU   $2A       ; [long] device's current mode
gdCCBytes       EQU   $2E       ; [word] depth of expanded cursor data
gdCCDepth       EQU   $30       ; [word] depth of expanded cursor data
gdCCXData       EQU   $32       ; [long] handle to cursor's expanded data
gdCCXMask       EQU   $36       ; [long] handle to cursor's expanded mask
gdReserved      EQU   $3A       ; [long] MUST BE 0

gdRec           EQU   $3E       ; size of GrafDevice record

; VALUES FOR GDType

clutType        EQU   0         ; 0 if lookup table
fixedType       EQU   1         ; 1 if fixed table
directType      EQU   2         ; 2 if direct values

;                   BIT ASSIGNMENTS FOR GDFlags

gdDevType       EQU   0         ; 0 = monochrome; 1 = color
ramInit         EQU   10        ; 1 if initialized from 'scrn' resource
mainScrn        EQU   11        ; 1 if main screen
allInit         EQU   12        ; 1 if all devices initialized
screenDevice    EQU   13        ; 1 if screen device [not used]
noDriver        EQU   14        ; 1 if no driver for this GDevice
scrnActive      EQU   15        ; 1 if in use

;                   CCrsr (Color Cursor) field offsets
;
;       NOTE THAT THE FIRST FOUR FIELDS ARE PARALLEL TO THE FIRST FOUR FIELDS
;       OF A PATTERN SO THAT PATCONVERT CAN BE USED TO EXPAND A CURSOR
;

crsrType        EQU   0             ;[WORD] CURSOR TYPE
crsrMap         EQU   crsrType+2    ;[LONG] HANDLE TO CURSOR'S PIXMAP
crsrData        EQU   crsrMap+4     ;[LONG] HANDLE TO CURSOR'S COLOR DATA
crsrXData       EQU   crsrData+4    ;[LONG] HANDLE TO EXPANDED DATA
crsrXValid      EQU   crsrXData+4   ;[WORD] DEPTH OF EXPANDED DATA (0 IF NONE)
crsrXHandle     EQU   crsrXValid+2  ;[LONG] HANDLE FOR FUTURE USE
crsr1Data       EQU   crsrXHandle+4 ;[16 WORDS] ONE-BIT DATA
crsrMask        EQU   crsr1Data+32  ;[16 WORDS] ONE-BIT MASK
```

```
crsrHotSpot     EQU   crsrMask+32     ;[POINT] HOT-SPOT FOR CURSOR
crsrXTable      EQU   crsrHotSpot+4   ;[LONG] TABLE ID FOR EXPANDED DATA
crsrID          EQU   crsrXTable+4    ;[LONG] ID FOR CURSOR

crsrRec         EQU   crsrID+4        ;SIZE OF CURSOR SAVE AREA
```

```
;                       CIcon (Color Icon) field offsets
;

iconPMap                EQU   0                     ;[PIXMAP] ICON'S PIXMAP
iconMask                EQU   iconPMap+pmRec        ;[BITMAP] 1-BIT VERSION OF ICON ONE-BIT
                              ; MASK FOR ICON
iconBMap                EQU   iconMask+bitmapRec    ;[BITMAP] 1-BIT VERSION OF ICON
iconData                EQU   iconBMap+bitmapRec    ;[LONG] HANDLE TO PIXMAP DATA
                                                    ;FOLLOWED BY BMAP AND MASK DATA
iconRec                 EQU   iconData+4            ;SIZE OF ICON HEADER

;
;                       Gamma Table format
;
gVersion                EQU   0                 ; [word] gamma version number
gType                   EQU   gVersion+2        ; [word] gamma data type
gFormulaSize            EQU   gType+2           ; [word] Formula data size
gChanCnt                EQU   gFormulaSize+2    ; [word] number of channels of data
gDataCnt                EQU   gChanCnt+2        ; [word] number of values/channel
gDataWidth              EQU   gDataCnt+2        ; [word] bits/corrected value (data packed to next
                                                ; larger byte size)
gFormulaData            EQU   gDataWidth+2      ; [array] data for formulas, followed by gamma values


;                       EXTENSIONS TO THE QDPROCS RECORD
;

opcodeProc              EQU   $34           ; [pointer]
newProc1                EQU    $38          ; [pointer]
newProc2                EQU   $3C           ; [pointer]
newProc3                EQU   $40           ; [pointer]
newProc4                EQU   $44           ; [pointer]
newProc5                EQU   $48           ; [pointer]
newProc6                EQU   $4C           ; [pointer]
cqdProcsRec             EQU   $50           ; size of QDProcs record

;                       OFFSETS WITHIN GRAFVARS:
;
rgbOpColor              EQU   0                 ; [6 bytes] color for addPin, subPin and average
rgbHiliteColor          EQU   rgbOpColor+6      ; [6 bytes] color for hiliting
pmFgColor               EQU   rgbHiliteColor+6  ; [4 bytes] palette handle for foreground color
pmFgIndex               EQU   pmFgColor+4       ; [2 bytes] index value for foreground
pmBkColor               EQU   pmFgIndex+2       ; [4 bytes] palette handle for background color
pmBkIndex               EQU   pmBkColor+4       ; [2 bytes] index value for background
pmFlags                 EQU   pmBkIndex+2       ; [2 bytes] flags for Palette Manager

grafVarRec              EQU   pmFlags+2         ; size of grafVar record


; color manager equates
```

```
; RGBColor structure

red             EQU   $0       ;red channel intensity      [short]
green           EQU   $2       ;green channel intensity    [short]
blue            EQU   $4       ;blue channel intensity     [short]
```

```
rgbColor            EQU   $6          ;size of record

; ColorSpec structure

value               EQU   $0          ;value field              [short]
rgb                 EQU   $2          ;rgb values               [rgbColor]
colorSpecSize       EQU   $8          ;size of record

; MatchRec structure

;red                EQU   $0          ;defined in RGBColor
;green              EQU   $2          ;defined in RGBColor
;blue               EQU   $4          ;defined in RGBColor
matchData           EQU   $6          ; [long]
matchRecSize        EQU   $A

;                   ColorTable field offsets
;

ctSeed              EQU   0           ; [long] id number for table
transIndex          EQU   ctSeed+4    ; [word] index of transparent pixel
ctSize              EQU   transIndex+2; [word] number of entries in CTTable
ctTable             EQU   ctSize+2    ; [variant] array of color spec

ctRec               EQU   ctTable     ; size of record without color table
ctEntrySize         EQU   8           ; size of each entry in table

minSeed             EQU   1023        ; minimum seed value (< minSeed reserved for rsrc ID's)

protect             EQU   7           ; protect bit is bit #8
reserveBit          EQU   6           ; reserve bit is bit #7

invalColReq         EQU   -1          ; invalid color table request

; CProcRec structure

nxtComp             EQU   $0          ;link to next proc     [pointer]
compProc            EQU   $4          ;pointer to routine    [pointer]
cProcSize           EQU   8           ; size of CProcRec

; inverse Table structure
ITabSeed            EQU   $0          ;[long] ID of owning color table
ITabRes             EQU   $4          ;[word] client ID
ITTable             EQU   $6          ;table of indices starts here

; SProcRec structure

nxtSrch             EQU   $0          ;[pointer] link to next proc
srchproc            EQU   $4          ;[pointer] pointer to routine
sProcSize           EQU   8           ; size of SProcRec
```

```
; request List structure

reqLSize        EQU   0         ; request list size [word]
reqLData        EQU   2         ; request list data [words]
```

;System Equates -- This file defines the low-level equates for the Macintosh software. The comments marked with ";+" denote categories or managers.

```
PCDeskPat       EQU   $20B      ; desktop pat, top bit only! others are in use
HiKeyLast       EQU   $216      ; Same as KbdVars
KbdLast         EQU   $218      ; Same as KbdVars+2

ExpandMem       EQU   $2B6      ; pointer to expanded memory block
expandSize      EQU   64        ; size of expanded memory block

; more specific fields for _Open

ioMix           EQU    $1C      ; General purpose field imported by driver[long]
ioFlags         EQU    $20      ; General purpose flags [word]
ioSlot          EQU    $22      ; Slot [byte]
ioId            EQU    $23      ; Id [byte]

ioSEBlkPtr      EQU    $22      ; Pointer to the seBlock [long]

; ioFlags:
fMulti          EQU   $00       ; b0 = fMulti: ioSEBlkPtr is valid (ioSlot, ioId are invalid)

; Test Manager EQU's
videoMagic      EQU   $5A932BC7; When VideoInfoOk contains this value, the video card is ok
(CritErr).


; Unit table size constants (Used in startInit.a)
unitEntries     EQU   64        ; default # of entries in unit table
maxUTEntries    EQU   unitEntries+64    ; Set Max higher so the table can grow.

bgnSlotUnit     EQU   48        ; default start unit number for slots.
bgnSlotRef      EQU   -(bgnSlotUnit+1)  ; default start RefNum for slots.



; Start Boot state constants.
sbState0        EQU   0         ;StartBoot code is at state-0.
sbState1        EQU   1         ;StartBoot code is at state-1.


; The Alarm Clock

alrmFlEnable    EQU   0         ; 1 => alarm clock mechanism is triggered

; start of new low mem

SCSIBase        EQU   $0C00     ; (long) base address for SCSI chip read
SCSIDMA         EQU   $0C04     ; (long) base address for SCSI DMA
```

```
SCSIHsk         EQU   $0C08     ; (long) base address for SCSI handshake
SCSIGlobals     EQU   $0C0C     ; (long) ptr for SCSI mgr locals

RGBBlack        EQU   $0C10     ; (6 bytes) the black field for color <C413>
RGBWhite        EQU   $0C16     ; (6 bytes) the white field for color <C413>

RowBits         EQU   $0C20     ; (word) screen horizontal pixels
```

```
ColLines        EQU   $0C22     ; (word) screen vertical pixels

ScreenBytes EQU   $0C24  ; (long) total screen bytes

; $0C28 unused (was SlotDT)

NMIFlag         EQU   $0C2C     ; (byte) flag for NMI debounce
VidType         EQU   $0C2D     ; (byte) video board type ID
VidMode         EQU   $0C2E     ; (byte) video mode (4=4bit color)

SCSIPoll        EQU   $0C2F     ; (byte) poll for device zero only once.
                                ; (init to $FFFF, by default)

SEVarBase       EQU   $0C30     ; ($0C30-0CB0) 128 bytes for sys err data
                                ; note!!! - if changed, need to change also in hwequ file

MMUFlags        EQU   $0CB0     ; (byte) cleared to zero (reserved for future use)
MMUType         EQU   $0CB1     ; (byte) kind of MMU present
MMU32bit        EQU   $0CB2     ; (byte) boolean reflecting current machine MMU mode
MMUFluff        EQU   $0CB3     ; (byte) fluff byte forced by reducing MMUMode to MMU32bit.
MMUTbl          EQU   $0CB4     ; (long) pointer to MMU Mapping table
MMUTblSize      EQU   $0CB8     ; (long) size of the MMU mapping table

SInfoPtr        EQU   $0CBC     ; (long) pointer to Slot manager information

ASCBase         EQU   $0CC0     ; (long) pointer to Sound Chip
SMGlobals       EQU   $0CC4     ; (long) pointer to Sound Manager Globals

TheGDevice      EQU   $0CC8     ; (long) the current graphics device
CQDGlobals      EQU   $0CCC     ; (long) quickDraw global extensions


; TEMPORARY EQUATES

DeskCPat        EQU   $0CD8     ;[PixPatHandle] Handle to desk pixPat
DeskPatDisable  EQU   $0CDC     ;[byte/boolean] if 0, then use deskCPat

ADBBase         EQU   $0CF8     ; (long) pointer to Front Desk Buss Variables

WarmStart       EQU   $0CFC     ; (long) flag to indicate it is a warm start
wmStConst       EQU   $574C5343 ; warm start constant

TimeDBRA        EQU   $0D00     ; (word) number of iterations of DBRA per millisecond
TimeSCCDB       EQU   $0D02     ; (word) number of iter's of SCC access & DBRA.

SlotQDT         EQU   $0D04     ; ptr to slot queue table
SlotPrTbl       EQU   $0D08     ; ptr to slot priority table
SlotVBLQ        EQU   $0D0C     ; ptr to slot VBL queue table
ScrnVBLPtr      EQU   $0D10     ; save for ptr to main screen VBL queue
SlotTICKS       EQU   $0D14     ; ptr to slot tickcount table
```

```
;4appletalk     EQU   $0D1C    ; (long) pointer to appletalk globals
TableSeed       EQU   $0D20    ; (long???) seed value for color table ID's
SRsrcTblPtr     EQU   $0D24    ; (long) pointer to slot resource table.
```

```
JVBLTask        EQU   $0D28     ; vector to slot VBL task interrupt handler

WMgrCPort       EQU   $0D2C     ; window manager color port

VertRRate       EQU   $0D30     ; (word) Vertical refresh rate for start manager.

; additional private low memory globals for nuMac's and later
;SynListHandle  EQU   $0D32     ;a handle to a list of synthesized fonts
;LastFore       EQU   $0D36     ;8 bytes: last foreground and background colors (FM)
;LastMode       EQU   $0D3E     ;word:   last text mode (Font Manager)
;LastDepth      EQU   $0D40     ;word:   last depth font prepared for
;FMExist        EQU   $0D42     ;byte:   clear if InitFonts has already been called

; Unused $0D43-0D53

;MBProcHndl     EQU   $0D54     ; Alladin/ handle to current menubar defproc
;mRect          EQU   $0D58     ; Alladin/ used by mbar defproc
;MenuCInfo      EQU   $0D5C     ; handle to menu color table

ChunkyDepth     EQU   $0D60     ; depth of the pixels
CrsrPtr         EQU   $0D62     ; pointer to cursor save area
PortList        EQU   $0D66     ; list of grafports<C14X>

MickeyBytes     EQU   $0D6A     ; long pointer to cursor stuff

QDErr           EQU   $0D6E     ; QuickDraw error code [word]

VIA2DT          EQU   $0D70     ; 32 bytes for VIA2 dispatch table for NuMac
                                ; uses $0D70 - $0D8F


SInitFlags      EQU   $0D90     ; StartInit.a flags [word]

DTQueue         EQU   $0D92     ; (10 bytes) deferred task queue header
DTQFlags        EQU   $0D92     ; flag word for DTQueue
DTskQHdr        EQU   $0D94     ; ptr to head of queue
DTskQTail       EQU   $0D98     ; ptr to tail of queue
JDTInstall      EQU   $0D9C     ; (long) ptr to deferred task install routine

HiliteRGB       EQU   $0DA0     ; 6 bytes:    rgb of hilite color

TimeSCSIDB      EQU   $0DA6     ; (word) number of iter's of SCSI access & DBRA

DSCtrAdj        EQU   $0DA8     ; (long) Center adjust for DS rect.

IconTLAddr      EQU   $0DAC     ; (long) pointer to where start icons are to be put.

VideoInfoOK     EQU   $0DB0     ; (long) Signals to CritErr that the Video card is ok

EndSRTPtr       EQU   $0DB4     ; (long) Pointer to the end of the Slot Resource Table (Not the
SRT buffer).
```

```
SDMJmpTblPtr    EQU    $0DB8    ; (long) Pointer to the SDM jump table

JSwapMMU        EQU    $0DBC    ; (long) jump vector to SwapMMU routine

SdmBusErr       EQU    $0DC0    ; (long) Pointer to the SDM busErr handler
LastTxGDevice   EQU    $0DC4    ; (long) copy of TheGDevice set up for fast text measure
```

```
; Unused $0DC8-...                          ; PLEASE MAINTAIN THIS POINTER TO UNUSED

; CRSRSAVE $88C-8CB is no longer used
; *** RESERVED FOR USE BY QUICKDRAW ***

NewCrsrJTbl             EQU     $88C        ; location of new crsr jump vectors
NewCrsrJCnt             EQU     1           ; 2 new vectors
JAllocCrsr              EQU     $88C        ; (long) vector to routine that allocates cursor
JSetCCrsr               EQU     $890        ; (long) vector to routine that sets color cursor
JOpcodeProc             EQU     $894        ; (long) vector to process new picture opcodes
CRSRBASE                EQU     $898        ; (long) scrnBase for cursor
CrsrDevice              EQU     $89C        ; (long) current cursor device
SrcDevice               EQU     $8A0        ; (LONG) Src device for Stretchbits
MainDevice              EQU     $8A4        ; (long) the main screen device
DeviceList              EQU     $8A8        ; (long) list of display devices
CRSRROW                 EQU     $8AC        ; (word) rowbytes for current cursor screen
QDColors                EQU     $8B0        ; (long) handle to default colors

; QuickDraw

HiliteMode              EQU     $938        ; used for color highlighting


; END OF TEMPORARY EQUATES


; Exception vectors

BusErrVct               EQU     $08         ; bus error vector


;-------------
; MMU Equates
;-------------

; MMU Mode bits
;
; type MMU_Mode = (true32b,false32b)

false32b                EQU     0           ;modified
true32b                 EQU     1

;+ System Error Handler

RestProc                EQU      $A8C        ; Resume procedure f        InitDialogs [pointer]

; equates for new queue elements

sIQType                 EQU     6           ; slot interrupt queue element ID       <C409>
```

```
;Default Startup

;DefaultRec offsets for set/get default startup

drDriveNum          EQU    0          ;[INTEGER]
drRefNum            EQU    2          ;[INTEGER]
```

```
; Deferred Task Queue Element

dtQType         EQU   7       ; deferred task queue element ID
inDTQ           EQU   6       ; bit index for "in deferred task" flag

dtLink          EQU   0       ; Link to next element [pointer]
dtType          EQU   4       ; Unique ID for validity [word]
dtFlags         EQU   6       ; optional flags [word]
dtAddr          EQU   8       ; service routine [pointer]
dtParm          EQU   $C      ; optional A1 parameter [long]
dtResrvd        EQU   $10     ; reserved [long]

dtQElSize       EQU   20      ; length of DT queue element in bytes


;+ ROM Equates

ROM85           EQU   $28E    ; (word) actually high bit - 0 for ROM vers $75 (sic) and later
ROMMapHndl      EQU   $B06    ; (long) handle of ROM resource map

;+ Screen Equates

ScrVRes         EQU   $102     ; screen vertical dots/inch [word]
ScrHRes         EQU   $104     ; screen horizontal dots/inch [word]
ScrnBase        EQU   $824     ; Screen Base [pointer]
ScreenRow       EQU   $106     ; rowBytes of screen [word]


; Mouse/Keyboard

MBTicks         EQU   $16E     ; tick count @ last mouse button [long]
JKybdTask       EQU   $21A     ; keyboard VBL task hook [pointer]
KeyLast         EQU   $184     ; ASCII for last valid keycode [word]
KeyTime         EQU   $186     ; tickcount when KEYLAST was rec'd [long]
KeyRepTime      EQU   $18A     ; tickcount when key was last repeated [long]

;+ Parameter RAM (a twenty byte copy of the real parameter ram).

SPConfig        EQU   $1FB     ; config bits: 4-7 A, 0-3 B (see use type below)
SPPortA         EQU   $1FC     ; SCC port A configuration [word]
SPPortB         EQU   $1FE     ; SCC port B configuration [word]

; SCC Serial Chip Addresses

SCCRd           EQU   $1D8     ; SCC base read address [pointer]
SCCWr           EQU   $1DC     ; SCC base write address [pointer]

; Serial port use type

useFree         EQU   0        ; Use undefined
useATalk        EQU   1        ; AppleTalk
```

```
useAsync        EQU     2       ; Async
useExtClk       EQU     3       ; externally clocked

; Unpacked, user versions of parameter ram

DoubleTime      EQU     $2F0    ; double click ticks [long]
```

```
CaretTime       EQU   $2F4      ; caret blink ticks [long]
KeyThresh       EQU   $18E      ; threshold for key repeat [word]
KeyRepThresh    EQU   $190      ; key repeat speed [word]
SdVolume        EQU   $260      ; Global volume(sound) control [byte]


;+ System Clocks

Ticks           EQU   $16A      ; Tick count, time since boot [long]
Time            EQU   $20C      ; clock time (extrapolated) [long]


;+ Cursor

iBeamCursor     EQU   1         ; text selection cursor
crossCursor     EQU   2         ; for structured selection
plusCursor      EQU   3         ; for drawing graphics
watchCursor     EQU   4         ; for indicating a long delay


; result codes for Relstring call

sortsBefore     EQU   -1        ; str1 < str2
sortsEqual      EQU   0         ; str1 = str2
sortsAfter      EQU   1         ; str1 > str2

;+ Queue Package

qInUse          EQU   7         ; queue-in-use flag bit

; Header Record

qHeadSize       EQU   $A        ; queue header size
qFlags          EQU   0         ; miscellaneous flags [word]
qHead           EQU   2         ; first element in queue [pointer]
qTail           EQU   6         ; last element in queue [pointer]

; General Purpose Queue Element Definition

qLink           EQU   0         ; link to next queue element [pointer]
qType           EQU   4         ; queue element type [word]


;+ Event Manager

evType          EQU   4         ; event queue element is type 4

; Event Type Numbers (in EvtNum)

nullEvt         EQU   0         ; event 0 is the null event
mButDwnEvt      EQU   1         ; mouse button down is event 1
```

```
mButUpEvt       EQU   2       ; mouse button up is event 2
keyDwnEvt       EQU   3       ; key down is event 3
keyUpEvt        EQU   4       ; key up is event 4
autoKeyEvt      EQU   5       ; auto-repeated key is event 5
updatEvt        EQU   6       ; update event
diskInsertEvt   EQU   7       ; disk-inserted event
activateEvt     EQU   8       ; activate/deactive event
```

```
netWorkEvt      EQU   $A        ; network event
ioDrvrEvt       EQU   $B        ; driver-defined event
app1Evt         EQU   $C        ; application defined events
app2Evt         EQU   $D
app3Evt         EQU   $E
app4Evt         EQU   $F

; Modifier bits in event record

activeFlag      EQU   $0        ; activate?
btnState        EQU   $7        ; state of button?
cmdKey          EQU   $8        ; command key down?
shiftKey        EQU   $9        ; shift key down?
alphaLock       EQU   $A        ; alpha lock down?
optionKey       EQU   $B        ; option key down?

; Event Record Definition

evtNum          EQU   0         ; event code [word]
evtMessage      EQU   2         ; event message [long]
evtTicks        EQU   6         ; ticks since startup [long]
evtMouse        EQU   $A        ; mouse location [long]
evtMeta         EQU   $E        ; state of modifier keys [byte]
evtMBut         EQU   $         ; state of mouse button [byte]
evtBlkSize      EQU   $10       ; size in bytes of the event record

MonkeyLives     EQU   $100      ; monkey lives if >= 0 [word]
SEvtEnb         EQU   $15C      ; enable SysEvent calls from GNE [byte]
JournalFlag     EQU   $8DE      ; journaling state [word]
JournalRef      EQU   $8E8      ; Journalling driver's refnum [word]


;+ Memory Manager

; Master pointer bits for handles - USE _HLock, _HPurge, etc. for portability

lock            EQU   7         ; lock bit in a master pointer
purge           EQU   6         ; bit for purgeable/unpurgeable
resource        EQU   5         ; bit to flag a resource handle

RSDHndl         EQU   $28A      ; resource driver handle (-1 until initialized)


BufPtr          EQU   $10C      ; top of application memory [pointer]
StkLowPt        EQU   $110      ; Lowest stack as measured in VBL task [pointer]
TheZone         EQU   $118      ; current heap zone [pointer]
ApplLimit       EQU   $130      ; application limit [pointer]
SysZone         EQU   $2A6      ; system heap zone [pointer]
ApplZone        EQU   $2AA      ; application heap zone [pointer]
HeapEnd         EQU   $114      ; end of heap [pointer]
HiHeapMark      EQU   $BAE      ; (long) highest address used by a zone below sp<01Nov85
```

JTC>

```
MemErr          EQU   $220          ; last memory manager error [word]

maxSize         EQU   $800000       ; outrageously large memory mgr request
dfltStackSize   EQU   $00002000     ; 8K size for stack
mnStackSize     EQU   $00000400     ; 1K minimum size for stack
```

```
; _InitZone argument table.

startPtr        EQU     0       ; Start address for zone [pointer]
limitPtr        EQU     4       ; Limit address for zone [pointer]
cMoreMasters    EQU     8       ; Number of masters to allocate at time [word]
pGrowZone       EQU     10      ; growZone procedure [pointer]

; Control/Status Call Codes

killCode        EQU     1       ; KillIO code
drvStsCode      EQU     8       ; status call code for drive status
ejectCode       EQU     7       ; control call eject code
tgBuffCode      EQU     8       ; set tag buffer code

; Driver flags, (Bit definitions for DCtlFlags byte)

dReadEnable     EQU     0       ; enabled for read operations
dWritEnable     EQU     1       ; enabled for writing
dCtlEnable      EQU     2       ; enabled for control operations
dStatEnable     EQU     3       ; enabled for status operations
dNeedGoodBye    EQU     4       ; needs a "goodbye kiss"
dNeedTime       EQU     5       ; needs "main thread" time
dNeedLock       EQU     6       ; needs to be accessed at interrupt level

; Run-Time flags, (Bit definitions for DCtlFlags+1 byte)

dOpened         EQU     5       ; bit to mark driver 'Open'
dRAMBased       EQU     6       ; 1=RAM-based Driver, 0=ROM-based
drvrActive      EQU     7       ; bit to mark the driver active

; Drive queue element offsets

dQDrive         EQU     $6      ; drive number [word]
dQRefNum        EQU     $8      ; driver refnum [word]
dQFSID          EQU     $A      ; file system handling this drive [word]
dQDrvSz         EQU     $C      ; number of blocks this drive [word]
dQDrvSz2        EQU     $E      ; if qType = 1, high word of drive size

; Queue Element Type Definitions

ioQType         EQU     2       ; I/O queue element is type 2
drvQType        EQU     3       ; timer queue element is type 3
fsQType         EQU     5       ; File System VCB element

; Device Control Entry Definition

dCtlEntrySize   EQU     $34 ; length of a DCE [52 bytes]
dCtlDriver      EQU     0       ; driver [handle]
dCtlFlags       EQU     4       ; flags [word]
dCtlQueue       EQU     6       ; queue header
```

```
dCtlQHead       EQU   8       ; queue first-element [pointer]
dCtlQTail       EQU   $C      ; queue last-element [pointer]
dCtlPosition    EQU   $10     ; position [long]
dCtlStorage     EQU   $14     ; driver's private storage [handle]
dCtlRefNum      EQU   $18     ; refNum of this driver [word]
dCtlCurTicks    EQU   $1A     ; counter for timing systemTask calls [long]
dCtlWindow      EQU   $1E     ; driver's window (if any) [pointer]
```

```
dCtlDelay       EQU     $22     ; number of ticks between sysTask calls [word]
dCtlEMask       EQU     $24     ; desk accessory event mask [word]
dCtlMenu        EQU     $26     ; menu ID associated with driver [word]
dCtlSlot        EQU     $28     ; device slot Number [byte]
dCtlSlotId      EQU     $29     ; device Id within slot [byte]
dCtlDevBase     EQU     $2A     ; driver scratch ptr/offset from base to device [long]
dCtlOwner       EQU     $2E     ; ptr to task control block(ownership) [Ptr]
dctlExtDev      EQU     $32     ; Id of external device [byte]


; Driver Globals

UTableBase      EQU     $11C    ; unit I/O table [pointer]
UnitNtryCnt     EQU     $1D2    ; count of entries in unit table [word]

JFetch          EQU     $8F4    ; fetch a byte routine for drivers [pointer]
JStash          EQU     $8F8    ; stash a byte routine for drivers [pointer]
JIODone         EQU     $8FC    ; IODone entry location [pointer]

;Chooser

chooserID       EQU     1       ; caller value for the chooser

;+ I/O System

; File positioning modes for ioPosMode field of I/O record

fsAtMark        EQU     0       ; at current position of mark
fsFromStart     EQU     1       ; offset relative to beginning of file
fsFromLEOF      EQU     2       ; offset relative to logical end-of-file
fsFromMark      EQU     3       ; offset relative to current mark
rdVerify        EQU     $40     ; read verify mode

; Permission values for ioPermssn field of I/O record

fsCurPerm       EQU     0       ; whatever is currently allowed
fsRdPerm        EQU     1       ; request to read only
fsWrPerm        EQU     2       ; request to write only
fsRdWrPerm      EQU     3       ; request to read and write
fsRdWrShPerm    EQU     4       ; request for shared read and write

; I/O record (general fields with trap-specific ones listed below)

ioQElSize       EQU     $32     ; length of I/O parameter block [50 bytes]

ioLink          EQU     $0      ; queue link in header [pointer]
ioType          EQU     $4      ; type for safety check [byte]
ioTrap          EQU     $6      ; the trap [word]
ioCmdAddr       EQU     $8      ; address to dispatch to [pointer]

ioCompletion    EQU     $C      ; completion routine [pointer]
```

```
ioResult        EQU   $10       ; I/O result code [word]
ioFileName      EQU   $12       ; file name pointer [pointer]
ioVRefNum       EQU   $16       ; volume refnum [word]
ioDrvNum        EQU   $16       ; drive number [word]
ioRefNum        EQU   $18       ; file reference number [word]

ioFileType      EQU   $1A       ; specified along with FileName [byte]
```

```
; specific fields for _Read, _Write

ioBuffer        EQU   $20       ; data buffer [pointer]
ioByteCount     EQU   $24       ; requested byte count [long]
ioNumDone       EQU   $28       ; actual byte count completed [long]

ioPosMode       EQU   $2C       ; initial file positioning mode/eol char [word]
ioPosOffset     EQU   $2E       ; file position offset [long]

; specific fields for _Allocate

ioReqCount      EQU   $24       ; requested new size [long]
ioActCount      EQU   $28       ; actual byte count allocated [long]

; specific fields for _Open

ioPermssn       EQU   $1B       ; permissions [byte]
ioOwnBuf        EQU   $1C       ; "private" 522-byte buffer [pointer]

; specific fields for _ReName

ioNewName       EQU   $1C       ; new name pointer [pointer]

; specific fields for     _GetFileInfo, _SetFileInfo

ioFQElSize      EQU   $50       ; File command parameter length [80 bytes]

ioFDirIndex     EQU   $1C       ; directory index of file [word]
ioFlAttrib      EQU   $1E       ; in-use bit=7, lock bit=0 [byte]
ioFFlType       EQU   $1F       ; file type [byte]
ioFlUsrWds      EQU   $20       ; finder info [16 bytes]
ioFFlNum        EQU   $30       ; file number [long]
ioDirID         EQU   $30       ; directory ID

ioFlStBlk       EQU   $34       ; start file block (0000 if none) [word]
ioFlLgLen       EQU   $36       ; logical length (EOF) [long]
ioFlPyLen       EQU   $3A       ; physical length in bytes [long]
ioFlRStBlk      EQU   $3E       ; resource fork's start file block [word]
ioFlRLgLen      EQU   $40       ; resource fork's logical length (EOF) [long]
ioFlRPyLen      EQU   $44       ; resource fork's physical length [long]

ioFlCrDat       EQU   $48       ; creation date & time [long]
ioFlMdDat       EQU   $4C       ; last modification date & time [long]

; Specific fields for _GetEOF, _SetEOF

ioLEOF          EQU   $1C       ; logical end-of-file [long]

; Specific fields for _SetFileType
```

```
ioNewType               EQU     $1C             ; new type byte [byte]

; Specific fields for _GetVolInfo, _GetVolume, _SetVolume, _MountVol, _UnmountVol,_Eject.
                                                ; Note that these traps have a bigger record size.

ioVQElSize              EQU     $40             ; Volume command parameter length [64 bytes]
ioVDrvNum               EQU     $16             ; drive or volume number [word]
```

```
ioVNPtr         EQU   $12       ; name buffer (or zero) [pointer]
ioVolIndex      EQU   $1C       ; volume index number [word]

ioVCrDate       EQU   $1E       ; creation date & time [long]
ioVLsBkUp       EQU   $22       ; last backup date & time [long]
ioVAtrb         EQU   $26       ; Volume attributes [word]
ioVNmFls        EQU   $28       ; # files in directory [word]
ioVDirSt        EQU   $2A       ; start block of file dir [word]
ioVBlLn         EQU   $2C       ; length of dir in blocks [word]
ioVNmAlBlks     EQU   $2E       ; num blks (of alloc size) this dev [word]
ioVAlBlkSiz     EQU   $30       ; alloc blk byte size [long]
ioVClpSiz       EQU   $34       ; bytes to try to allocate at a time [long]
ioAlBlSt        EQU   $38       ; starting block in block map [word]
ioVNxtFNum      EQU   $3A       ; next free file number [long]
ioVFrBlk        EQU   $3E       ; # free alloc blks for this vol [word]

;--- New File System Equates    ---
;
; Catalog structure equates:

fsRtParID       EQU   1         ; DirID of parent's root
fsRtDirID       EQU   2         ; Root DirID
fsXTCNID        EQU   3         ; Extent B*-Tree file ID
fsCTCNID        EQU   4         ; Catalog B*-Tree file ID
fsUsrCNID       EQU   $10       ; First assignable user CNode ID

; Additional equates for catalog information return:

ioFlBkDat       EQU   $50       ; File's last backup date
ioFlxFndrInfo   EQU   $54       ; File's additional finder info bytes
ioFlParID       EQU   $64       ; File's parent directory ID
ioFlClpSiz      EQU   $68       ; File's clump size, in bytes

; Additional equates for directory information return:

ioDirFlg        EQU   4         ; Bit in ioFlAttrb set to indicate directory
ioDrUsrWds      EQU    $20      ; Directory's user info bytes
ioDrDirID       EQU   $30       ; Directory ID
ioDrNmFls       EQU   $34       ; Number of files in a directory
ioDrCrDat       EQU   $48       ; Directory creation date
ioDrMdDat       EQU   $4C       ; Directory modification date
ioDrBkDat       EQU   $50       ; Directory backup date
ioDrFndrInfo    EQU   $54       ; Directory finder info bytes
ioDrParID       EQU   $64       ; Directory's parent directory ID

ioHFQElSiz      EQU    $6C      ; Size of a Hierarchical File Queue Element

; Additional equates for _TFGetVolInfo:

ioVLsMod        EQU   $22       ; Last modification date
ioVSigWord      EQU   $40       ; Volume signature
```

```
ioVCBVBMst      EQU   $2A
ioVNxtCNID      EQU   $3A
ioVDrvInfo      EQU   $42     ; Drive number (0 if volume is offline)
ioVDRefNum      EQU   $44     ; Driver refNum
ioVFSID         EQU   $46     ; ID of file system handling this volume
ioVBkup         EQU   $48     ; Last backup date (0 if never backed up)
ioVSeqNum       EQU   $4C     ; Sequence number of this volume in volume set
```

```
ioVWrCnt        EQU   $4E      ; Volume write count
ioVFilCnt       EQU   $52      ; Total number of files on volume
ioVDirCnt       EQU   $56      ; Total number of directories on the volume
ioVFndrInfo EQU       $5A      ; Finder information for volume

ioHVQElSize EQU       $7A      ; Length of Hierarchical Volume information PB

; New fields for _GetFCBInfo:

ioFCBIndx       EQU   $1C       ; FCB index for _GetFCBInfo
ioFCBFiller1    EQU   $1E       ; filler
ioFCBFlNm       EQU   $20       ; File number
ioFCBFlags      EQU   $24       ; FCB flags
ioFCBStBlk      EQU   $26       ; File start block
ioFCBEOF        EQU   $28       ; Logical end-of-file
ioFCBPLen       EQU   $2C       ; Physical end-of-file
ioFCBCrPs       EQU   $30       ; Current file position
ioFCBVRefNum    EQU   $34       ; Volume refNum
ioFCBClpSiz     EQU   $36       ; File clump size
ioFCBParID      EQU   $3A       ; Parent directory ID

; New fields for _GetWDInfo:

ioWDIndex       EQU   $1A       ; Working Directory index for _GetWDInfo
ioWDProcID      EQU   $1C       ; WD's ProcID (long)
ioWDVRefNum     EQU   $20       ; WD's Volume RefNum (word)
ioWDDirID       EQU   $30       ; WD's DirID (long)

; New fields for _FSControl call:

ioFSVrsn        EQU   $20       ; File system version

; New field for CatMove

ioNewDirID      EQU    $24      ;destination directory for CatMove

;
;--- End of New File System Equates ---

; Specific fields for _Control, _Status

csCode          EQU   $1A       ; control/status code [word]
csParam         EQU   $1C       ; operation-defined parameters [22 bytes]

; FInfo (Finder Information) record layout

fdType          EQU   $0        ; type of file [long]
fdCreator       EQU   $4        ; file's creator [long]
fdFlags         EQU   $8        ; flags [word]
fdLocation      EQU   $A        ; file's location [point]
fdFldr          EQU   $E        ; file's window [word]
```

```
; added for HFS

; FXInfo record layout

fdIconID        EQU     $0      ; Icon ID [word]
fdUnused        EQU     $2      ; unused but reserved [4 words]
```

```
fdComment       EQU   $A          ; Comment ID [word]
fdPutAway       EQU   $C          ; Home Dir ID [2 words]

; DInfo record layout

frRect          EQU   $0          ; Folder Rect [4 words]
frFlags         EQU   $8          ; Flags [word]
frLocation      EQU   $A          ; Location [2 words]
frView          EQU   $E          ; Folder view [word]

; DXInfo record layout

frScroll        EQU   $0          ; scroll position [2 words]
frOpenChain     EQU   $4          ; dirID chain of open folders [2 words]
frUnused        EQU   $8          ; unused but reserved [word]
frComment       EQU   $A          ; comment [word]
frPutAway       EQU   $C          ; Dir ID [2 words]

;end of addition

; Masks for fdFlags field of FInfo record defined above

fHasBundle      EQU   13           ; set if file has a bundle
fInvisible      EQU   14           ; set if file's icon is invisible
fTrash          EQU   -3           ; file is in Trash window
fDeskTop        EQU   -2           ; file is on desktop
fDisk           EQU   0            ; file is in disk window

; File System Globals

DrvQHdr         EQU   $308         ; queue header of drives in system [10 bytes]
BootDrive       EQU   $210         ; drive number of boot drive [word]
EjectNotify     EQU   $338         ; eject notify procedure [pointer]
IAZNotify       EQU   $33C         ; world swaps notify procedure [pointer]
SFSaveDisk      EQU   $214         ; last vRefNum seen by standard file [word]
CurDirStore     EQU   46+$36A      ; save dir across calls to Standard File [long]


;+ Date-Time record (for use with _Secs2Date, and _Date2Secs)


dtYear          EQU   $0           ; year (1904..) [word]
dtMonth         EQU   $2           ; month (1..12) [word]
dtDay           EQU   $4           ; day (1..31) [word]
dtHour          EQU   $6           ; hour (0..23) [word]
dtMinute        EQU   $8           ; minute (0..59) [word]
dtSecond        EQU   $A           ; second (0..59) [word]
dtDayOfWeek     EQU   $C           ; day of week, sunday..saturday (1..7) [word]


;+ Miscellaneous stuff
```

```
OneOne          EQU   $A02      ; constant $00010001 [long]
MinusOne        EQU   $A06      ; constant $FFFFFFFF [long]
Lo3Bytes        EQU   $31A      ; constant $00FFFFFF [long]

ROMBase         EQU   $2AE      ; ROM base address [pointer]
```

```
RAMBase         EQU   $2B2      ; RAM base address [pointer]
SysVersion      EQU   $15A      ; version # of RAM-based system [word]
RndSeed         EQU   $156      ; random seed/number [long]

;+ Scratch Areas

scratch20       EQU   $1E4      ; scratch [20 bytes]
scratch8        EQU   $9FA      ; scratch [8 bytes]

;+ Scrap Manager

ScrapSize       EQU   $960      ; scrap length [long]
ScrapHandle     EQU   $964      ; memory scrap [handle]
ScrapCount      EQU   $968      ; validation byte [word]
ScrapState      EQU   $96A      ; scrap state [word]
ScrapName       EQU   $96C      ; pointer to scrap name [pointer]

;+ Desk Accessories

; Message Definitions (in CSCode of control call)

accEvent        EQU   $40       ; event message from SystemEvent
accRun          EQU   $41       ; run message from SystemTask
accCursor       EQU   $42       ; cursor message from SystemTask
accMenu         EQU   $43       ; menu message from SystemMenu
accUndo         EQU   $44       ; undo message from SystemEdit
accCut          EQU   $46       ; cut message from SystemEdit
accCopy         EQU   $47       ; copy message from SystemEdit
accPaste        EQU   $48       ; paste message from SystemEdit
accClear        EQU   $49       ; clear message from SystemEdit

goodBye         EQU   -1        ; goodbye message

;International stuff

IntlSpec        EQU   $BA0      ; (long) - ptr to extra Intl data

;Switcher

SwitcherTPtr EQU    $286    ; Switcher's switch table

; Trap bits for memory manager

tSysOrCurZone   EQU   10        ; bit set implies System Zone
```

```
                                        ; bit clear implies Current Zone
clearBit        EQU     9               ; bit set means clear allocated memory.

; Peripheral chips and Magic Hardware addresses
```

```
CPUFlag         EQU   $12F      ; $00=68000, $01=68010, $02=68020 (old ROM inits to $00)
                                ; (this is old DskWr11 flag . . .)

; VIA (6522) interface chip

VIA             EQU   $1D4      ; VIA base address [pointer]

; Disk Address

IWM             EQU   $1E0      ; IWM base address [pointer]


; Interrupt secondary vectors

Lvl1DT          EQU   $192      ; Interrupt level 1 dispatch table [32 bytes]
Lvl2DT          EQU   $1B2      ; Interrupt level 2 dispatch table [32 bytes]
ExtStsDT        EQU   $2BE      ; SCC ext/sts secondary dispatch table [16 bytes]



; Parameter Ram

SPValid         EQU   $1F8      ; validation field ($A7) [byte]
SPATalkA        EQU   $1F9      ; AppleTalk node number hint for port A
SPATalkB        EQU   $1FA      ; AppleTalk node number hint for port B
SPAlarm         EQU   $200      ; alarm time [long]
SPFont          EQU   $204      ; default application font number minus 1 [word]
SPKbd           EQU   $206      ; kbd repeat thresh in 4/60ths [2 4-bit]

SPPrint         EQU   $207      ; print stuff [byte]
SPVolCtl        EQU   $208      ; volume control [byte]
SPClikCaret     EQU   $209      ; double click/caret time in 4/60ths[2 4-bit]

SPMisc1         EQU   $20A      ; miscellaneous [1 byte]
SPMisc2         EQU   $20B      ; miscellaneous [1 byte]

GetParam        EQU   $1E4      ; system parameter scratch [20 bytes]
SysParam        EQU   $1F8      ; system parameter memory [20 bytes]

; Cursor

CrsrThresh      EQU   $8EC      ; delta threshold for mouse scaling [word]
JCrsrTask       EQU   $8EE      ; address of CrsrVBLTask [long]
MTemp           EQU   $828      ; Low-level interrupt mouse location [long]
RawMouse        EQU   $82C      ; un-jerked mouse coordinates [long]
CrsrRect        EQU   $83C      ; Cursor hit rectangle [8 bytes]
TheCrsr         EQU   $844      ; Cursor data, mask & hotspot [68 bytes]
CrsrAddr        EQU   $888      ; Address of data under cursor [long]
CrsrSave        EQU   $88C      ; data under the cursor [64 bytes]
CrsrVis         EQU   $8CC      ; Cursor visible? [byte]
CrsrBusy        EQU   $8CD      ; Cursor locked out? [byte]
```

```
CrsrNew         EQU   $8CE      ; Cursor changed? [byte]
CrsrState       EQU   $8D0      ; Cursor nesting level [word]
CrsrObscure     EQU   $8D2      ; Cursor obscure semaphore [byte]
```

```
; Mouse/Keyboard

KbdVars         EQU   $216     ; Keyboard manager variables [4 bytes]
KbdType         EQU   $21E     ; keyboard model number [byte]
MBState         EQU   $172     ; current mouse button state [byte]
KeyMap          EQU   $174     ; bitmap of the keyboard [2 longs]
KeypadMap       EQU   $17C     ; bitmap for numeric pad-18bits [long]
Key1Trans       EQU   $29E     ; keyboard translator procedure [pointer]
Key2Trans       EQU   $2A2     ; numeric keypad translator procedure [pointer]
JGNEFilter      EQU   $29A     ; GetNextEvent filter proc [pointer]
KeyMVars        EQU   $B04     ; (word) for ROM KEYM proc state

Mouse           EQU   $830     ; processed mouse coordinate [long]
CrsrPin         EQU   $834     ; cursor pinning rectangle [8 bytes]
CrsrCouple      EQU   $8CF     ; cursor coupled to mouse? [byte]
CrsrScale       EQU   $8D3     ; cursor scaled? [byte]
MouseMask       EQU   $8D6     ; V-H mask for ANDing with mouse [long]
MouseOffset     EQU   $8DA     ; V-H offset for adding after ANDing [long]

; System Clocks

AlarmState      EQU   $21F     ; Bit7=parity, Bit6=beeped, Bit0=enable [byte]


;+ Vertical Blanking Interrupt Handler

; VBL Block Queue Element

vType           EQU   1        ; VBL queue element is type 1
inVbl           EQU   6        ; bit index for "in VBL" flag

vblink          EQU   0        ; Link to next element [pointer]
vblType         EQU   4        ; Unique ID for validity [word]
vblAddr         EQU   6        ; service routine [pointer]
vblCount        EQU   $A       ; timeout count [word]
vblPhase        EQU   $C       ; phase count [word]
VBLQueue        EQU   $160     ; VBL queue header [10 bytes]

; Event manager

jPlayCtl        EQU   16       ; playBack call
jRecordCtl      EQU   17       ; record call
jcTickCount     EQU   0        ; journal code for TickCount
jcGetMouse      EQU   1        ; journal code for GetMouse
jcButton        EQU   2        ; journal code for Button
jcGetKeys       EQU   3        ; journal code for GetKeys
jcEvent         EQU   4        ; journal code for GetNextEvent(Avail)

SysEvtMask      EQU   $144     ; system event mask [word]
SysEvtBuf       EQU   $146     ; system event queue element buffer [pointer]
EventQueue      EQU   $14A     ; event queue header [10 bytes]
```

```
EvtBufCnt       EQU   $154      ; max number of events in SysEvtBuf - 1 [word]

; Event Queue Element Data Stucture

evtQWhat        EQU   6         ; event code [word]
evtQMessage     EQU   8         ; event message [long]
evtQWhen        EQU   $C        ; ticks since startup [long]
```

```
evtQWhere       EQU   $10       ; mouse location [long]
evtQMeta        EQU   $14       ; state of modifier keys [byte]
evtQMBut        EQU   $15       ; state of mouse button [byte]
evtQBlkSize     EQU   $16       ; size of event record counting queue info

; flags in flags field in heapzone header

fOnCheck        EQU   0         ; Turn On Checking
fChecking       EQU   1         ; Checking on
fNSelCompct     EQU   4         ; Use non-selective compact algorithm when 1.
fNoRvrAlloc     EQU   5         ; Don't use rover allocation scheme when 1.
fNSelPurge      EQU   6         ; Use non-selective purge algorithm when 1.
fRelAtEnd       EQU   7         ; MakeBk packs rels at end of free bk when 1.

ROZ             EQU   $0        ; bit in flags field of MemMgr zone header

; Block Types

tybkMask        EQU   3         ; Mask for block type
tybkFree        EQU   0         ; Free Block
tybkNRel        EQU   1         ; Non-Relocatable
tybkRel         EQU   2         ; Relocatable

; Block Offsets

tagBC           EQU   0         ; Tag and Byte Count field [long]
handle          EQU   4         ; back pointer to master pointer [pointer]
blkData         EQU   8         ; data starts here

; Heap Zone header

bkLim           EQU   $0        ; last block in zone [pointer]
purgePtr        EQU   $4        ; roving purge placeholder [pointer]
hFstFree        EQU   $8        ; first free handle [pointer]
zcbFree         EQU   $C        ; # of free bytes in zone [long]
gzProc          EQU   $10       ; grow zone procedure [pointer]
mAllocCnt       EQU   $14       ; # of master ptrs to allocate [word]
flags           EQU   $16       ; Flags [word]
cntRel          EQU   $18       ; # of allocated relocatable blocks [word]
maxRel          EQU   $1A       ; max # of allocated rel. blocks [word]
cntNRel         EQU   $1C       ; # of allocated non-rel. blocks [word]
maxNRel         EQU   $1E       ; max # of allocated non-rel. blocks [word]
cntEmpty        EQU   $20       ; # of empty handles [word]
cntHandles      EQU   $22       ; total # of handles [word]
minCBFree       EQU   $24       ; min # of bytes free [long]
purgeProc       EQU   $28       ; purge warning procedure [pointer]
allocPtr        EQU   $30       ; roving allocator [pointer]
heapData        EQU   $34       ; start of heap zone data
```

```
GZRootHnd       EQU   $328      ; root handle for GrowZone [handle]
GZRootPtr       EQU   $32C      ; root pointer for GrowZone [pointer]
GZMoveHnd       EQU   $330      ; moving handle for GrowZone [handle]
MemTop          EQU   $108      ; top of memory [pointer]
MmInOK          EQU   $12E      ; initial memory mgr checks ok? [byte]
HpChk           EQU   $316      ; heap check RAM code [pointer]
MaskBC          EQU   $31A      ; Memory Manager Byte Count Mask [long]
```

```
MaskHandle      EQU   $31A      ; Memory Manager Handle Mask [long]
MaskPtr         EQU   $31A      ; Memory Manager Pointer Mask [long]
MinStack        EQU   $31E      ; min stack size used in InitApplZone [long]
DefltStack      EQU   $322      ; default size of stack [long]
MMDefFlags      EQU   $326      ; default zone flags [word]

;+ System Error Handler

DSAlertTab      EQU   $2BA      ; system error alerts [pointer]
DSAlertRect     EQU   $3F8      ; rectangle for disk-switch alert [8 bytes]
DSDrawProc      EQU   $334      ; alternate syserror draw procedure [pointer]
DSWndUpdate     EQU   $15D      ; GNE not to paintBehind DS AlertRect? [byte]
WWExist         EQU   $8F2      ; window manager initialized? [byte]
QDExist         EQU   $8F3      ; quickdraw is initialized [byte]
ResumeProc      EQU   $A8C      ; Resume procedure from InitDialogs [pointer]
DSErrCode       EQU   $AF0      ; last system error alert ID

;+ Drivers

dskRfN          EQU   $FFFB     ; 3.5" disk reference number
IntFlag         EQU   $15F      ; reduce interrupt disable time when bit 7 = 0

; Serial I/O Driver

SerialVars      EQU   $2D0      ; async driver variables [16 bytes]

PortAUse        EQU   $290      ; bit 7: 1 = not in use, 0 = in use
                                ; bits 0-3: current use of port (see use type)
                                ; bits 4-6: user specific
PortBUse        EQU   $291      ; port B use, same format as PortAUse
SCCASts         EQU   $2CE      ; SCC read reg 0 last ext/sts rupt - A [byte]
SCCBSts         EQU   $2CF      ; SCC read reg 0 last ext/sts rupt - B [byte]

; Serial handshake record definition

shFXOn          EQU   $0       ; XOn/XOff output control flags [byte]
shFCTS          EQU   $1       ; CTS hardware handshake flag [byte]
shXOn           EQU   $2       ; XOn character [byte]
shXOff          EQU   $3       ; XOff character [byte]
shErrs          EQU   $4       ; errors that cause abort [byte]
shEvts          EQU   $5       ; status changes that cause events [byte]
shFInX          EQU   $6       ; XOn/XOff input flow control flag [byte]
shNull          EQU   $7       ; not used [byte]

; Serial status record definition

ssCumErrs       EQU   $0       ; cumulative errors [byte]
ssXOffSent      EQU   $1       ; XOff sent as input control flag [byte]
ssRdPend        EQU   $2       ; read pending flag [byte]
ssWrPend        EQU   $3       ; write pending flag [byte]
ssCTSHold       EQU   $4       ; CTS flow control hold flag [byte]
```

```
ssXOffHold              EQU     $5              ; XOff received as output flow control [byte]
; Disk Driver
; Driver Code Header (for I/O drivers, desk accessories)
drvrFlags               EQU     $0              ; various flags and permissions [word]
```

```
drvrDelay       EQU   $2        ; # of ticks between systask calls [word]
drvrEMask       EQU   $4        ; event mask [word]
drvrMenu        EQU   $6        ; driver menu ID [word]
drvrOpen        EQU   $8        ; open routine offset [word]
drvrPrime       EQU   $A        ; prime routine offset [word]
drvrCtl         EQU   $C        ; control routine offset [word]
drvrStatus      EQU   $E        ; status routine offset [word]
drvrClose       EQU   $10        ; warmstart reset routine offset [word]
drvrName        EQU   $12        ; length byte and name of driver [string]

; Driver Status record definition

dsTrack         EQU   $0        ; current track [word]
dsWriteProt     EQU   $2        ; bit 7=1 if volume locked [byte]
dsDiskInPlace   EQU   $3        ; disk in place [byte]
dsInstalled     EQU   $4        ; drive installed [byte]
dsSides         EQU   $5        ; bit 7=0 if single-sided drive [byte]
dsQLink         EQU   $6        ; next queue entry [pointer]
dsDQVers        EQU   $A        ; 1 for HD20 [word]
dsDQDrive       EQU   $C        ; drive number [word]
dsDQRefNum      EQU   $E        ; driver reference number [word]
dsDQFSID        EQU   $10        ; file-system identifier [word]
dsTwoSideFmt    EQU   $12        ; -1 if two-sided disk [byte]
dsDiskErrs      EQU   $14        ; error count [word]

dsDrvSize       EQU $12         ;drive block size low word [word]
dsDrvS1         EQU $14         ;drive block size high word [word]
dsDrvType       EQU $16         ;1 for HD20 [word]
dsDrvManf       EQU $18         ;1 for Apple Computer, Inc [word]
dsDrvChar       EQU $1A         ;230 ($E6) for HD20 [word]
dsDrvMisc       EQU $1C         ;0 -- reserved [byte]


DskErr          EQU   $142       ; disk routine result code [word]
PWMBuf2         EQU   $312       ; PWM buffer 1 (or 2 if sound) [pointer]

; Drive command codes

dcRead          EQU   0
dcWrite         EQU   1
dcStatus        EQU   3
dcInit          EQU   25
dcScan          EQU   26

; Sound Stuff

SoundPtr        EQU   $262       ; 4VE sound definition table [pointer]
SoundBase       EQU   $266       ; sound bitMap [pointer]
SoundVBL        EQU   $26A       ; vertical retrace control element [16 bytes]
SoundDCE        EQU   $27A       ; sound driver DCE [pointer]
SoundActive     EQU   $27E       ; sound is active? [byte]
```

```
SoundLevel      EQU     $27F        ; current level in buffer [byte]
CurPitch        EQU     $280        ; current pitch value [word]

;               I/O System

noQueueBit      EQU     $9          ; tells I/O system not to queue the request
asyncTrpBit     EQU     $A          ; bit in high byte of trap specifying async
```

```
toExtFS             EQU   $3F2        ; hook for external file systems

; File System Globals

DskVerify           EQU   $12C        ; used by 3.5 disk driver for read/verify [byte]
TagData             EQU   $2FA        ; sector tag info for disk drivers [14 bytes]
BufTgFNum           EQU   $2FC        ; file number [long]
BufTgFFlg           EQU   $300        ; flags [word]
BufTgFBkNum         EQU   $302        ; logical block number [word]
BufTgDate           EQU   $304        ; time stamp [word]

; I/O Command Equates for I/O Queue Elements (match trap numbers)

aRdCmd              EQU   2           ; read command
aWrCmd              EQU   3           ; write command
aCtlCmd             EQU   4           ; control command
aStsCmd             EQU   5           ; status command

; New fields for _SetPMSP call: PMSP = "Poor Man's Search Path"

ioPMSPFlg           EQU   $1A         ; Flag whether to enable the PMSP
ioPMSPHook          EQU    $1C        ; Pointer to PMSP hook proc

; Print variables

ScrDmpEnb           EQU   $2F8        ; screen dump enabled? [byte]
ScrDmpType          EQU   $2F9        ; FF dumps screen, FE dumps front window [byte]

; Scrap Variables

ScrapVars           EQU   $960        ; scrap manager variables [32 bytes]
ScrapInfo           EQU   $960        ; scrap length [long]
ScrapEnd            EQU   $980        ; end of scrap vars
ScrapTag            EQU   $970        ; scrap file name [STRING[15]]

; Segment Loader

LaunchFlag          EQU   $902        ; from launch or chain [byte]
SaveSegHandle       EQU   $930        ; seg 0 handle [handle]
CurJTOffset         EQU   $934        ; current jump table offset [word]
CurPageOption       EQU   $936        ; current page 2 configuration [word]
LoaderPBlock        EQU   $93A        ; param block for ExitToShell [10 bytes]
CurApRefNum         EQU   $900        ; refNum of application's resFile [word]
CurrentA5           EQU   $904        ; current value of A5 [pointer]
CurStackBase        EQU   $908        ; current stack base [pointer]
CurApName           EQU   $910        ; name of application [STRING[31]]
LoadTrap            EQU   $12D        ; trap before launch? [byte]

SegHiEnable         EQU   $BB2        ; (byte) 0 to disable MoveHHi in LoadSeg
```

```
;device manager - Chooser message values

newSelMsg       EQU   12      ;a new selection has been made
fillListMsg     EQU   13      ;fill the list with choices to be made
getSelMsg       EQU   14      ;mark one or more choices as selcted
selectMsg       EQU   15      ;a choice has actually been made
deselectMsg     EQU   16      ;a choice has been canceled
```

```
terminateMsg    EQU   17      ;lets device package clean up
buttonMsg       EQU   19      ;a button has been clicked

psAlert         EQU   6       ;page setup alert bit in HiliteMode

theChooser      EQU   1
```

```
Toolbox Equates -- This file defines the high-level equates for the
                   Macintosh toolbox software.  The comments marked with ";+" denote managers.


;+ Resource Manager

; Resource attributes

resSysRef       EQU   7       ; reference to system/local reference
resSysHeap      EQU   6       ; In system/in application heap
resPurgeable    EQU   5       ; Purgeable/not purgeable
resLocked       EQU   4       ; Locked/not locked
resProtected    EQU   3       ; Protected/not protected
resPreload      EQU   2       ; Read in at OpenResource?
resChanged      EQU   1       ; Existing resource changed since last update

rcbMask         EQU   $FD     ; Must preserve ResChanged over _ResAttrs

; Map attributes

mapReadOnly     EQU   7       ; is this file read-only?
mapCompact      EQU   6       ; Is a compact necessary?
mapChanged      EQU   5       ; Is it necessary to write map?

; Resource Manager Globals

TopMapHndl      EQU   $A50    ; topmost map in list [handle]
SysMapHndl      EQU   $A54    ; system map [handle]
SysMap          EQU   $A58    ; reference number of system map [word]
CurMap          EQU   $A5A    ; reference number of current map [word]
ResReadOnly     EQU   $A5C    ; Read only flag [word]
ResLoad         EQU   $A5E    ; Auto-load feature [word]
ResErr          EQU   $A60    ; Resource error code [word]
ResErrProc      EQU   $AF2    ; Resource error procedure [pointer]
SysResName      EQU   $AD8    ; Name of system resource file [STRING[19]]

;new Resource Manager stuff

RomMapInsert    EQU   $B9E    ; (byte) determines if we should link in map
TmpResLoad      EQU    $B9F   ; second byte is temporary ResLoad value.

; the following word values are to be placed into the
; word located at RomMapInsert

MapTrue         EQU   $FFFF   ; link in ROM map with resload true
MapFalse        EQU   $FF00   ; link in ROM map with resload false

;+ Font Manager


; Standard font ID's
```

```
sysFont         EQU   0         ; system font ID
applFont        EQU   1         ; application font ID

newYork         EQU   2         ; standard release fonts
geneva          EQU   3
```

```
monaco          EQU     4
venice          EQU     5
london          EQU     6
athens          EQU     7
sanFran         EQU     8
toronto         EQU     9
cairo           EQU     11
losAngeles      EQU     12
times           EQU     20
helvetica       EQU     21
courier         EQU     22
symbol          EQU     23
mobile          EQU     24

; Font Manager Globals

ApFontID        EQU     $984        ; resource ID of application font [word]
FMDefaultSize   EQU     $987        ; default size [byte]
CurFMInput      EQU     $988        ; quickdraw FMInput Record [pointer]
FMgrOutRec      EQU     $998        ; quickdraw FontOutput Record [pointer]
FScaleDisable   EQU     $A63        ; disable font scaling? [byte]

;new FONT manager stuff

WidthListHand   EQU     $8E4       ; list of extra width tables, or nil.
WidthPtr        EQU     $B10       ; (long) Font Mgr global
WidthTabHandle  EQU     $B2A       ; Font width table handle for measure
LastSPExtra     EQU     $B4C       ; (long) most recent value of space extra
SysFontFam      EQU     $BA6       ; (word) System font family ID or zero
SysFontSize     EQU     $BA8       ; (word) System font size (or zero for 12 pt)
FDevDisable     EQU     $BB3       ; (byte) $FF to disable device-defined style extra
LastFOND        EQU     $BC2       ; (long) handle of last font def record
FONDID          EQU     $BC6       ; (word) ID of last font def record
FractEnable     EQU     $BF4       ; (byte) flag for fractional font widths
UsedFWidths     EQU     $BF5       ; (byte) flag saying if we used fract widths
FScaleHFact     EQU     $BF6       ; (long) horz. font scale factor
FScaleVFact     EQU     $BFA       ; (long) vertical font scale factor

;+ Window Manager

                                   ; system windows have negative kinds
dialogKind      EQU     2          ; dialog windows
userKind        EQU     8          ; this and above numbers are for user

; Values returned by window definition function's hit routine

wNoHit          EQU     0          ; not in window at all
wInContent      EQU     1          ; in content area
wInDrag         EQU     2          ; in drag area
wInGrow         EQU     3          ; in grow area
wInGoAway       EQU     4          ; in go away area
```

```
wInZoomIn       EQU   5       ; in zoom in
wInZoomOut      EQU   6       ; in zoom out

; FindWindow Return Codes

inDesk          EQU   0       ; not in any window
inMenuBar       EQU   1       ; in the menu bar
```

```
inSysWindow     EQU  2          ; in a system window
inContent       EQU  3          ; in content area of user window
inDrag          EQU  4          ; in drag area of user window
inGrow          EQU  5          ; in grow area of user window
inGoAway        EQU  6          ; in go away area of user window
inZoomIn        EQU  7          ; in zoom in part code
inZoomOut       EQU  8          ; in zoom out part code

; Resource ID's for windows

deskPatID       EQU  16         ; desk pattern PAT ID

documentProc    EQU  0          ; standard document WDEF ID
dBoxProc        EQU  1          ; dialog box (document without titleBar) WDEF ID
plainDBox       EQU  2          ; no border WDEF ID
altDBoxProc     EQU  3          ; no shadow or title WDEF ID
noGrowDocProc   EQU  4          ; no grow area WDEF ID
zoomDocProc     EQU  8          ; with zoom box WDEF ID
zoomNoGrow      EQU 12 ; zoom with no grow box WDEF ID
rDocProc        EQU 16 ; document with rounded corners WDEF ID

; Window Data Structure Definition

windowPort      EQU  0          ; grafPort [108 bytes]
windowKind      EQU  $6C        ; type of window [word]
wVisible        EQU  $6E        ; visible flag [byte]
wHilited        EQU  $6F        ; select (hilite) flag [byte]
wGoAway         EQU  $70        ; has go away button [byte]
wZoom           EQU  $71        ; has zoom box [byte]
structRgn       EQU  $72        ; structure region of window [Handle]
contRgn         EQU  $76        ; content region of window [Handle]
updateRgn       EQU  $7A        ; update region of window [Handle]
windowDef       EQU  $7E        ; window definition procedure [Handle]
wDataHandle     EQU  $82        ; window proc-defined data [Handle]
wTitleHandle    EQU  $86        ; title string [Handle]
wTitleWidth     EQU  $8A        ; width in pixels of title string [word]
wControlList    EQU  $8C        ; control list of this window [handle]
nextWindow      EQU  $90        ; next window in z-ordered list [pointer]
windowPic       EQU  $94        ; picture handle for updates [handle]
wRefCon         EQU  $98        ; application use [long]
windowSize      EQU  $9C        ; size of window data structure

; Window Manager Globals

WindowList      EQU  $9D6       ; Z-ordered linked list of windows [pointer]
PaintWhite      EQU  $9DC       ; erase newly drawn windows? [word]
WMgrPort        EQU  $9DE       ; window manager's grafport [pointer]
GrayRgn         EQU  $9EE       ; rounded gray desk region [handle]
CurActivate     EQU  $A64       ; window slated for activate event [pointer]
CurDeactive     EQU  $A68       ; window slated for deactivate event [pointer]
```

```
DragHook        EQU   $9F6     ; user hook during dragging [pointer]
DeskPattern     EQU   $A3C     ; desk pattern [8 bytes]
DeskHook        EQU   $A6C     ; hook for painting the desk [pointer]
GhostWindow     EQU   $A84     ; window hidden from FrontWindow [pointer]
```

```
;+ Menu Manager

;                       "ASCII" marks for menu characters

noMark          EQU     0
commandMark     EQU     $11         ; command fan (cloverleaf)
checkMark       EQU     $12         ; check mark for menus
diamondMark     EQU     $13         ; diamond mark for menus
appleMark       EQU     $14         ; desk ornament menu title

; MenuList Data Structure Definition -- one per menuBar

                                    ; 6 Byte header
lastMenu        EQU     0           ; number of bytes in this menuList [word]
lastRight       EQU     2           ; h coordinate of 1st free point in menuBar [word]
                                    ; one of the following per menu
menuoH          EQU     0           ; menu handle [handle]
menuLeft        EQU     4           ; coordinate of left edge of menu [word]

; MenuInfo Data Structure -- one per menu

menuID          EQU     0           ; unique ID for each menuBar [word]
menuWidth       EQU     2           ; menu width [word]
menuHeight      EQU     4           ; menu height [word]
menuDefHandle   EQU     6           ; menu definition proc [handle]
menuEnable      EQU     $A          ; enable flags, one bit/item [long]
menuData        EQU     $E          ; menu item string [STRING]
menuBlkSize     EQU     $E          ; size of a menu block plus dataString

; MenuString Data Structure -- one per menu item

itemIcon        EQU     0           ; icon byte
itemCmd         EQU     1           ; apple (command key) byte
itemMark        EQU     2           ; checkmark character byte
itemStyle       EQU     3           ; style byte

; Menu Manager Globals

MenuList        EQU     $A1C        ; current menuBar list structure [handle]
MenuFlash       EQU     $A24        ; flash feedback count [word]
MenuHook        EQU     $A30        ; user hook during menuSelect [pointer]
MBarEnable      EQU     $A20        ; menuBar enable for desk accessories[word]
MBarHook        EQU     $A2C        ; user hook during menuSelect [pointer]

;new Menu Manager stuff

MBarHeight      EQU      $BAA       ; (word) height of menu bar (usually 20)

;+ Control Manager

; Part Codes
```

```
inButton        EQU   10      ; in a push button
inCheckBox      EQU   11      ; in a checkBox button
inUpButton      EQU   20      ; in up button area of a dial
inDownButton    EQU   21      ; in down button area of a dial
inPageUp        EQU   22      ; in page up (gray) area of a dial
inPageDown      EQU   23      ; in page down (gray) area of a dial
```

```
inThumb             EQU   129          ; in thumb area of a dial

; Constants for axis parameter of DragGrayRgn and DragControl

noConstraint        EQU   0            ; free form dragging
hAxisOnly           EQU   1            ; horizontally only
vAxisOnly           EQU   2            ; vertically only

; Resource ID's for controls

pushButProc         EQU   0            ; rounded-corner pushButtons CDEF ID
checkBoxProc        EQU   1            ; check-box type buttons CDEF ID
radioButProc        EQU   2            ; radio buttons      CDEF ID
scrollBarProc       EQU   16           ; scrollBar CDEF ID
useWFont            EQU   8            ; add this to get window font CDEF ID

sBarPatID           EQU   17           ; scrollBar pattern ID

; Control Template

nextControl         EQU   $0           ; next control in the list [handle]
contrlOwner         EQU   $4           ; owning window [pointer]
contrlRect          EQU   $8           ; bounding rectangle [8 bytes]
contrlVis           EQU   $10          ; visible state [byte]
contrlHilite        EQU   $11          ; hilite state [byte]
contrlValue         EQU   $12          ; current value of control [word]
contrlMin           EQU   $14          ; minimum value of control [word]
contrlMax           EQU   $16          ; maximum value of control [word]
contrlDefHandle     EQU   $18          ; control definition procedure [handle]
contrlData          EQU   $1C          ; data for definition proc [handle]
contrlAction        EQU   $20          ; local actionProc [pointer]
contrlRFcon         EQU   $24          ; refcon defined by application [long]
contrlTitle         EQU   $28          ; title string [STRING]
contrlSize          EQU   $28          ; size of control data structure less title

; Control Manager Globals

DragPattern         EQU   $A34         ; DragTheRgn pattern [8 bytes]
DragFlag            EQU   $A44         ; implicit parameter to DragControl [word]
CurDragAction       EQU   $A46         ; implicit actionProc for dragControl [pointer]


;+ Text Edit


; Justification styles

teJustLeft          EQU   0            ; left justified text
teJustRight         EQU   -1           ; right justified text
teJustCenter        EQU   1            ; center justified text
teForceLeft         EQU    -2          ; for Arabic fonts, force left justification
```

```
; Text Edit Record

teDestRect      EQU   $0       ; destination rectangle       [8 bytes]
teViewRect      EQU   $8       ; view rectangle rectangle [8 bytes]

teSelRect       EQU   $10       ; select rectangle [8 bytes]
```

```
teLineHite      EQU   $18     ; lineheight [word]
teAscent        EQU   $1A     ; first baseline offset [word]
teSelPoint      EQU   $1C     ; selection point [long]

teSelStart      EQU   $20     ; selection start [word]
teSelEnd        EQU   $22     ; selection end [word]

teActive        EQU   $24     ; active [byte]

teWordBreak     EQU   $26     ; word break routine [pointer]
teClikProc      EQU   $2A     ; click loop routine [pointer]

teClikTime      EQU   $2E     ; time of last click [long]
teClikLoc       EQU   $32     ; location of double click [long]

teCarTime       EQU   $34     ; time for next caret toggle [long]
teCarOn         EQU   $38     ; is caret on? [byte]
teCarAct        EQU   $39     ; is caret active? [byte]
teJust          EQU   $3A     ; fill style [word]

teLength        EQU   $3C     ; length of text below [word]
teTextH         EQU   $3E     ; text [handle]

teRecBack       EQU   $42     ; unused [word]
teRecLine       EQU   $44     ; unused [word]
teLftClick      EQU   $46     ; click was to left? [byte]
teLftCaret      EQU   $47     ; caret was to left? [byte]

teCROnly        EQU   $48     ; <CR> only for line breaks? [byte]

teFontStuff     EQU   $4A     ; space for font specifier [8 bytes]
teFont          EQU   $4A     ; text font [word]
teFace          EQU   $4C     ; text face [word]
teMode          EQU   $4E     ; text mode [word]
teSize          EQU   $50     ; text size [word]
teGrafPort      EQU   $52     ; grafport for editting [pointer]

teHiHook        EQU   $56     ; hook for hilite routine [pointer]
teCarHook       EQU   $5A     ; hook for hilite routine [pointer]

teNLines        EQU   $5E     ; number of lines [word]
teLines         EQU   $60     ; line starts [words...]

teRecSize       EQU   $68     ; base size of a record w/o lines

; Text Edit Globals

TEScrpLength    EQU   $AB0    ; textEdit Scrap Length [word]
TEScrpHandle    EQU   $AB4    ; textEdit Scrap [handle]
TEWdBreak       EQU   $AF6    ; default word break routine [pointer]
```

```
;new TE stuff

WordRedraw      EQU   $BA5        ; (byte) - used by TextEdit RecalDraw
TESysJust       EQU   $BAC        ; (word) system justification (intl. textEdit)
TEFlags         EQU   teRecBack   ; turn whole byte into bit flags
teFAutoPos      EQU   6           ; set this bit for auto position/scroll
```

```
;+ Dialog Manager

; Item codes in item list

userItem        EQU   0         ; application-defined (dialog only)
ctrlItem        EQU   4         ; must be added to following four items
btnCtrl         EQU   0         ; standard button
chkCtrl         EQU   1         ; standard check box
radCtrl         EQU   2         ; standard radio button
resCtrl         EQU   3         ; control defined in resource file
statText        EQU   8         ; static text
editText        EQU   16        ; editable text (dialog only)
iconItem        EQU   32        ; icon
picItem         EQU   64        ; quickdraw picture
itemDisable     EQU   128       ; add to any of above to disable


; Generic buttons

okButton        EQU   1         ; OK button
cancelButton    EQU   2         ; Cancel button

; Alert/Dialog Resource ID's

stopIcon        EQU   0         ; stop icon ID
noteIcon        EQU   1         ; note icon ID
cautionIcon     EQU   2         ; caution icon ID

; Dialog Template

dBounds         EQU   $0        ; dialog bounds rectangle
dWindProc       EQU   $8        ; window proc ID
dVisible        EQU   $A        ; visible flag
dGoAway         EQU   $C        ; go away flag
dRefCon         EQU   $E        ; reference constant
dItems          EQU   $12       ; item list ID and handle
dTitle          EQU   $14       ; dialog window title

; Alert Template

aBounds         EQU   $0        ; alert box height and width
aItems          EQU   $8        ; item list ID
aStages         EQU   $A        ; stages word

; Dialog/Alert Window Record

dWindow         EQU   $0        ; window record
items           EQU   $9C       ; Item list [handle]
teHandle        EQU   $A0       ; textEdit object [handle]
editField       EQU   $A4       ; current field being edited [word]
editOpen        EQU   $A6       ; is editting open? [word]
```

```
aDefItem        EQU   $A8       ; default item for alerts [word]
dWindLen        EQU   $AA       ; dialog record length

; In each item

itmHndl         EQU   0         ; handle to the item
itmRect         EQU   $4        ; bounding rect of item
```

```
itmType         EQU   $C        ; item type
itmData         EQU   $D        ; item string, must be even length

; Dialog Manager Globals

ANumber         EQU   $A98      ; active alert ID [word]
ACount          EQU   $A9A      ; # times this alert called [word]
DABeeper        EQU   $A9C      ; beep routine [pointer]
DAStrings       EQU   $AA0      ; paramText substutution strings [4 handles]
DlgFont         EQU   $AFA      ; default dialog font ID [word]


;+ Package Globals


AppPacks        EQU   $AB8      ; packages' code [8 handles]


;+ Finder related Globals


FinderName      EQU   $2E0      ; "Finder" name [STRING[15]]
AppParmHandle   EQU   $AEC      ; handle to hold application parameters


;+ Miscellaneous Globals


ApplScratch     EQU   $A78      ; application scratch area [12 Bytes]
ToolScratch     EQU   $9CE      ; scratch area [8 bytes]
TempRect        EQU   $9FA      ; scratch rectangle [8 bytes]

; System Patterns

sysPatListID    EQU   0          ; ID of PAT# which contains 38 patterns

; Resource Manager

mCCMask         EQU   $60       ; mapCompact + mapChanged
mChMask         EQU   $20       ; mapChanged
mCoMask         EQU   $40       ; mapCompact

; Font Manager

; Font header values

propFont        EQU   $9000     ; proportional font type
prpFntH         EQU   $9001     ; with height table
prpFntW         EQU   $9002     ; with width table
prpFntHW        EQU   $9003     ; with height & width table
```

```
fixedFont       EQU   $B000     ; fixed-pitch font type
fxdFntH         EQU   $B001     ; with height table
fxdFntW         EQU   $B002     ; with width table
fxdFntHW        EQU   $B003     ; with height & width table

fontWid         EQU   $ACB0     ; width-only font type
```

```
; control/status codes for linkage w/font manager

fMgrCtl1        EQU     8           ; printer drivers

; Font Header Data Record

fFontType       EQU     0           ; font type [word]
fFirstChar      EQU     2           ; ASCII code of first char [word]
fLastChar       EQU     4           ; ASCII code of last char [word]
fWidMax         EQU     6           ; maximum width of any char in pixels [word]
fKernMax        EQU     8           ; Negative of maximum character kern [word]
fNDescent       EQU     10          ; negative of descent [word]
fFRectWidth     EQU     12          ; width of font rectangle [word]
fFRectHeight    EQU     14          ; height of font rectangle [word]
fOWTLoc         EQU     16          ; offset to offset/width table [word]
fAscent         EQU     18          ; ascent above baseline in pixels [word]
fDescent        EQU     20          ; descent below baseline in pixels [word]
fLeading        EQU     22          ; space between lines in pixels [word]
fRowWords       EQU     24          ; row width of bit image / 2 [word]

; Font Manager Input Record (CurFMInput)

fmInFamily      EQU     0           ; family [word]
fmInSize        EQU     2           ; size [word]
fmInFace        EQU     4           ; face [word]
fmInNeedBits    EQU     5           ; needBits [byte]
fmInDevice      EQU     6           ; device number [byte]
fmInNumer       EQU     8           ; numerator of scale [fixed]
fmInDenom       EQU     12          ; denominator of scale [fixed]


; Font Manager Output record (FMgrOutRec)

fmOutError      EQU     0           ; error code [word]
fmOutFontH      EQU     2           ; the actual font [handle]
fmOutBold       EQU     6           ; bolding factor [byte]
fmOutItalic     EQU     7           ; italic factor [byte]
fmOutULOffset   EQU     8           ; underline offset [byte]
fmOutULShadow   EQU     9           ; underline halo [byte]
fmOutULThick    EQU     10          ; underline thickness [byte]
fmOutShadow     EQU     11          ; shadow factor [byte]
fmOutExtra      EQU     12          ; extra horizontal width [byte]
fmOutAscent     EQU     13          ; height above baseline [byte]
fmOutDescent    EQU     14          ; height below baseline [byte]
fmOutWidMax     EQU     15          ; maximum width of character [byte]
fmOutLeading    EQU     16          ; space between lines [byte]
fmOutNumer      EQU     18          ; point for numerators of scale factor [long]
fmOutDenom      EQU     22          ; point for denominators of scale factor [long]


;WidthTable data structure
```

```
widTabData      EQU     0       ;ARRAY[1..256] OF LONGINT character widths
widTabFont      EQU     1024    ;Handle         font record used to build table
widthSExtra     EQU     1028    ;LONGINT space extra used for table
widthStyle      EQU     1032    ;LONGINT extra due to style
widthFID        EQU     1036    ;INTEGER font family ID
widthFSize      EQU     1038    ;INTEGER font size request
```

```
widthFace       EQU   1040    ;INTEGER style (face) request
widthDevice     EQU   1042    ;INTEGER device requested
widthVInScale   EQU   1044    ;FIXED scale factors requested
widthHInScale   EQU   1048    ;FIXED scale factors requested
widthAFID       EQU   1052    ;INTEGER actual font family ID for table
widthFHand      EQU   1054    ;Handle family record used to build up table
widthUsedFam    EQU   1058    ;BOOLEAN used fixed point family widths
widthAFace      EQU   1059    ;BYTE actual face produced
widthVOutput    EQU   1060    ;INTEGER vertical scale output value
widthHOutput    EQU   1062    ;INTEGER horizontal scale output value
widthVFactor    EQU   1064    ;INTEGER vertical scale output value
widthHFactor    EQU   1066    ;INTEGER horizontal scale output value
widthASize      EQU   1068    ;INTEGER actual size of actual font used
widTabSize      EQU   1070    ;INTEGER total size of table

; Font Family Definition

ffFlags         EQU   0       ; flags for family (word)
ffFamID         EQU   2       ; family ID number (word)
ffFirst         EQU   4       ; ASCII code of first character (word)
ffLast          EQU   6       ; ASCII code of last character (word)
ffAscent        EQU   8       ; maximum ascent expressed for 1 pt (word)
ffDescent       EQU   10      ; maximum descent expressed for 1 pt (word)
ffLeading       EQU   12      ; maximum leading expressed for 1 pt (word)
ffWidMax        EQU   14      ; maximum widMax expressed for 1 pt (word)
ffWTabOff       EQU   16      ; offset to width table (long)
ffKernOff       EQU   20      ; offset to kerning table (long)
ffStylOff       EQU   24      ; offset to style mapping table (long)
ffProperty      EQU   28      ; style property info (12 words)
ffIntl          EQU   52      ; reserved for international use (2 words)
ffVersion       EQU   56      ; FOND version number

; Font Characterization Table

dpiVert         EQU   0       ; vertical dots per inch [word]
dpiHoriz        EQU   2       ; horizontal dots per inch [word]
boldChr         EQU   4       ; bold characteristics [3 bytes]
italChr         EQU   7       ; italic characteristics [3 bytes]
                              ; unused [3 bytes]
outlineChr      EQU   13      ; outline characteristics [3 bytes]
shadowChr       EQU   16      ; shadow characteristics [3 bytes]
condChr         EQU   19      ; condensed characteristics [3 bytes]
extendChr       EQU   22      ; extended characteristics [3 bytes]
underChr        EQU   25      ; underline characteristics [3 bytes]

; Globals

CurFMFamily     EQU   $988     ; current font family
CurFMSize       EQU   $98A     ; current font size
CurFMFace       EQU   $98C     ; current font face
CurFMNeedBits   EQU   $98D     ; boolean specifying whether it needs strike
```

```
CurFMDevice     EQU   $98E     ; current font device
CurFMNumer      EQU   $990     ; current numerator of scale factor
CurFMDenom      EQU   $994     ; current denominator of scale factor

FOutRec         EQU   $998     ; Font Manager output record

FMDotsPerInch   EQU   $9B2     ; h,v dotsPerInch of current device
```

```
FMStyleTab      EQU   $9B6      ; style heuristic table supplied by device

RomFont0        EQU   $980      ; system font [handle]
GotStrike       EQU   $986      ; Do we have the strike? [byte]


; Window Manager

; Window Definition Procedure Messages

wDrawMsg        EQU   0         ; draw yourself
wHitMsg         EQU   1         ; hit test
wCalcRgnMsg     EQU   2         ; recalculate your regions
wInitMsg        EQU   3         ; initialize yourself
wDisposeMsg     EQU   4         ; dispose any private data
wGrowMsg        EQU   5         ; drag out grow outline
wGIconMsg       EQU   6         ; draw the grow icon

OldStructure    EQU   $9E6      ; saved structure region [handle]
OldContent      EQU   $9EA      ; saved content region [handle]
SaveVisRgn      EQU   $9F2      ; temporarily saved visRegion [handle]
CurDeKind       EQU   $A22      ; window kind of deactivated window [word]
SaveUpdate      EQU   $9DA      ; Enable update accumulation? [word]

; Menu Manager

; Menu Definition Procedure Messages

mDrawMsg        EQU   0         ; draw yourself
mChooseMsg      EQU   1         ; select an item
mSizeMsg        EQU   2         ; calculate your size

; Menu Resource IDs

textMenuProc    EQU   0         ; standard text menu MDEF ID

maxMenu         EQU   $60       ; maximum of 16*6 menus in menuBar
mListSize       EQU   $66       ; menu list is 102 bytes long

TheMenu         EQU   $A26      ; ID of hilited menu [word]
SavedHandle     EQU   $A28      ; saved bits under a menu [handle]

;misc Menu stuff

MrMacHook       EQU   $A2C      ; Mr. Macintosh hook [pointer]

; Control manager

; Control Definition Procedure Messages

drawCtlMsg      EQU   0         ; draw message
```

```
hitCtlMsg       EQU  1      ; hit test message
calcCtlMsg      EQU  2      ; calc region message
newCtlMsg       EQU  3      ; init message
dispCtlMsg      EQU  4      ; dispose any private data message
posCtlMsg       EQU  5      ; adjust indicator position message
thumbCtlMsg     EQU  6      ; calculate rectangles for thumb dragging
dragCtlMsg      EQU  7      ; custom drag message
```

```
trackCtlMsg             EQU     8               ; track yourself message

; Text Edit

TEDoText                EQU     $A70            ; textEdit doText proc hook [pointer]
TERecal                 EQU     $A74            ; textEdit recalText proc hook [pointer]

;stage definition--packed 2 to a byte, 4 stages in a word

volBits                 EQU     3               ; number of beeps
alBit                   EQU     4               ; alert bit (put up box this time?)
okDismissal             EQU     8               ; bit for OK/Cancel default in each stage

; DialogList Data Structure Definitions

dlgMaxIndex             EQU     0               ; maximum index (=items-1) stored here


SaveProc                EQU     $A90            ; address of Save failsafe procedure
SaveSP                  EQU     $A94            ; Safe SP for restart or save

; Package Manager

FPState                 EQU     $A4A            ; floating point state [6 bytes]
App2Packs               EQU     $BC8           ; $BC8-$BE7 eight more package handles

; Resource Manager

RMGRPerm                EQU     $BA4            ; (byte) - permission byte for OpenResFile

; Miscellaneous Constants

screenRadius            EQU     $00100010 ; rounded corners for desk area

; Miscellaneous Globals

IconBitmap              EQU     $A0E            ; bitmap used for plotting things
TaskLock                EQU     $A62            ; re-entering SystemTask [byte]
CloseOrnHook            EQU     $A88            ; hook for closing desk ornaments

;new MacApp stuff

MAErrProc               EQU     $BE8            ; (long) MacApp error proc address
MASuperTab              EQU     $BEC            ; (long) handle to MacApp superclass table


;**************** NEW TOOL EQUATES ************

; Font Manager
```

```
; addition to FMgrOutRec (was unused)
fmOutCurStyle           EQU     17              ;style algorthmically applied by QuickDraw
;_____________________________________________________________________________
;
```

```
; Window Manager

; auxWinRec structure

awNext          EQU   $0        ;next in chain          [Handle]
awOwner         EQU   $4        ;owner ID               [WindowPtr]
awCTable        EQU   $8        ;color table            [CTabHandle]
dialogCTable    EQU   $C        ;handle to dialog manager structures    [handle]
awFlags         EQU   $10       ;handle for Ernie       [handle]
awResrv         EQU   $14       ;for expansion          [longint]
awRefCon        EQU   $18       ;user constant          [longint]
auxWinSize      EQU   $1C       ;size of record

AuxWinHead      EQU   $0CD0     ;[handle] Window Aux List head

; Window Part Identifiers which correlate color table entries with window elements

wContentColor   EQU   0
wFrameColor     EQU   1
wTextColor      EQU   2
wHiliteColor    EQU   3
wTitleBarColor  EQU   4


;_____________________________________________________________________________
;
; Control Manager

; auxCtlRec structure

acNext          EQU   $0        ;next in chain          [AuxCtlHndl]
acOwner         EQU   $4        ;owner ID               [ControlHandle]
acCTable        EQU   $8        ;color table            [CCTabHandle]
acFlags         EQU   $C        ;misc flag byte         [word]
acReserved      EQU   $E        ;for expansion          [LONGINT]
acRefCon        EQU   $12       ;user constant          [LONGINT]
acSize          EQU   $16       ;size of record

AuxCtlHead      EQU   $0CD4     ;[handle] Control Aux List head

;               Here are some equates for the colors of control parts

cFrameColor     EQU   0
cBodyColor      EQU   1
cTextColor      EQU   2
cThumbColor     EQU   3

;_____________________________________________________________________________
;
; Menu Manager
```

```
MenuDisable             EQU     $0B54       ; menuID and Item when disabled item selected
MBDFHndl                EQU     $0B58       ; handle to current menu bar defproc
MBSaveLoc               EQU     $0B5C       ; handle to the mbarproc private data
MenuCInfo               EQU     $0D50       ; hanel to menu color information table

; Leftover Alladdin ROM equates
MBProcHndl              EQU     $0D54       ; handle to current menubar defproc
```

```
MRect                   EQU     $0D58       ; used by Alladin's mbar proc
MBFlash                 EQU     $0D5C       ; used by Alladin's mbar proc

; The following two equates have never been defined in an equate file, they were in
; the mdefproc.         The locations $B26 and $B26 were orginally used, and built in to
; the MacPlus and Alladdin Roms, but since scrolling had to work on 64K ROM machines
; $A0A and $A0C were chosen for that.       Hence forth the following values will be used.

TopMenuItem             EQU     $A0A        ; pixel value of top of scrollable menu
AtMenuBottom            EQU     $A0C        ; pixel value of bottom of scrollable menu

;
; color menu table equates (mct = menu color table)
;
mctID                   EQU     $0
mctItem                 EQU     $2
mctRGB1                 EQU     $4
mctRGB2                 EQU     $A
mctRGB3                 EQU     $10
mctRGB4                 EQU     $16
mctReserved             EQU     $1C
mctEntrySize            EQU     $1E

;
; miscellaneous equates for hierarchical menus
;

hMenuCmd                EQU     $1B         ; itemCmd == $1B ==> hierarchical menu for this
hierMenu                EQU     -1          ; InsertMenu(handle, hierMenu), when beforeID ==
                                            ; hierMenu, the handle is inserted in the
                                            ; hierarchical menuList
mPopUpMsg               EQU     4           ; menu defProc messages

menuDelay               EQU     $7E         ; param ram locations for user settable
menuDrag                EQU     $7F         ;  hierarchical menu delay and drag ticks

;
; miscellaneous menubar equates
;
mbMenu1Loc              EQU     $A          ; first menu is 10 pixels from left side of screen

;
; color menu table search (and destroy) messages (mct = menu color table)
;
mctAllIds               EQU     -97         ; search for all IDs for the given Item
mctAllItems             EQU     -98         ; search for all Items for the given ID
mctLastIDIndic          EQU     -99         ; last entry in color table has this in ID field


;_____________________________________________________________________________
;
```

```
;                       Background Notification Manager

BNMQHd                  EQU     $B50        ; head of background notification Q

;_____________________________________________________________________________
;
```

```
; Text Edit

; Set/Replace style modes

fontBit         EQU   0         ; set font
faceBit         EQU   1         ; set face
sizeBit         EQU   2         ; set size
clrBit          EQU   3         ; set color
addSizeBit      EQU   4         ; add size mode

; handle to style record

teStylesH       EQU   $4A       ; replaces teFont/teFace

; offsets into TEStyleRec

nRuns           EQU   0         ; [INTEGER] # of entries in styleStarts array
nStyles         EQU   2         ; [INTEGER] # of distinct styles
styleTab        EQU   4         ; [STHandle] handle to distinct styles
lhTab           EQU   8         ; [LHHandle] handle to line heights
teRefCon        EQU   12        ; [LONGINT] reserved
teReserved      EQU   16        ; [LONGINT] reserved
runs            EQU   20        ; array of styles


; offsets into StyleRun array

startChar       EQU   0         ; [INTEGER] offset into text to start of style
styleIndex      EQU   2         ; [INTEGER] style index

stStartSize     EQU   4         ; size of a styleStarts entry

; offsets into STElement

stCount         EQU   0         ; [INTEGER] # of times this style is used
stHeight        EQU   2         ; [INTEGER] line height
stAscent        EQU   4         ; [INTEGER] ascent
stFont          EQU   6         ; [INTEGER] font
stFace          EQU   8         ; [Style] face
stSize          EQU   10        ; [INTEGER] size
stColor         EQU   12        ; [RGBColor] color

stRecSize       EQU   18        ; size of a teStylesRec      ** <C182/6oct86/MBK> **

; offsets into TextStyle

tsFont          EQU   0         ; [INTEGER] font
tsFace          EQU   2         ; [Style] face
tsSize          EQU   4         ; [INTEGER] size
tsColor         EQU   6         ; [RGBColor] color
```

```
styleSize               EQU     12          ; size of a StylRec             ** <C182/6oct86/MBK> **

; offsets into StScrpRec

scrpNStyles             EQU     0           ; [INTEGER] # of styles in scrap
scrpStyleTab            EQU     2           ; [ScrpSTTable] start of scrap styles array
```

```
; offsets into scrpSTElement

scrpStartChar       EQU   0       ; [LONGINT] char where this style starts
scrpHeight          EQU   4       ; [INTEGER] line height
scrpAscent          EQU   6       ; [INTEGER]ascent
scrpFont            EQU   8       ; [INTEGER]font
scrpFace            EQU   10      ; [Style] face
scrpSize            EQU   12      ; [INTEGER]size
scrpColor           EQU   14      ; [RGBColor] color

scrpRecSize         EQU   20      ; size of a scrap record
```

```
; System Error Equates -- This file defines the equates for the Macintosh return error codes

  ; General System Errors (VBL Mgr, Queueing, Etc.)

  noErr          EQU  0          ; 0 for success
  qErr           EQU  -1         ; queue element not found during deletion
  vTypErr        EQU  -2         ; invalid queue element
  corErr         EQU  -3         ; core routine number out of range
  unimpErr       EQU  -4         ; unimplemented core routine
  seNoDB         EQU  -8         ; no debugger installed to handle debugger command <what
  num???>

  ; I/O System Errors

  controlErr     EQU  -17
  statusErr      EQU  -18
  readErr        EQU  -19
  writErr        EQU  -20
  badUnitErr     EQU  -21
  unitEmptyErr   EQU  -22
  openErr        EQU  -23
  closErr        EQU  -24
  dRemovErr      EQU  -25        ; tried to remove an open driver
  dInstErr       EQU  -26        ; DrvrInstall couldn't find driver in resources
  abortErr       EQU  -27        ; IO call aborted by KillIO
  notOpenErr     EQU  -28        ; Couldn't rd/wr/ctl/sts cause driver not opened

  ;              File System error codes:

  dirFulErr      EQU  -33        ; Directory full
  dskFulErr      EQU  -34        ; disk full
  nsvErr         EQU  -35        ; no such volume
  ioErr          EQU  -36        ; I/O error (bummers)
  bdNamErr       EQU  -37        ; there may be no bad names in the final system!
  fnOpnErr       EQU  -38        ; File not open
  eofErr         EQU  -39        ; End of file
  posErr         EQU  -40        ; tried to position to before start of file (r/w)
  mFulErr        EQU  -41        ; memory full (open) or file won't fit (load)
  tmfoErr        EQU  -42        ; too many files open
  fnfErr         EQU  -43        ; File not found

  wPrErr         EQU  -44        ; diskette is write protected
  fLckdErr       EQU  -45        ; file is locked
  vLckdErr       EQU  -46        ; volume is locked
  fBsyErr        EQU  -47        ; File is busy (delete)
  dupFNErr       EQU  -48        ; duplicate filename (rename)
  opWrErr        EQU  -49        ; file already open with with write permission
  paramErr       EQU  -50        ; error in user parameter list
  rfNumErr       EQU  -51        ; refnum error
  gfpErr         EQU  -52        ; get file position error
  volOffLinErr   EQU  -53        ; volume not on line error (was Ejected)
```

```
permErr         EQU   -54     ; permissions error (on file open)
volOnLinErr     EQU   -55     ; drive volume already on-line at MountVol
nsDrvErr        EQU   -56     ; no such drive (tried to mount a bad drive num)
noMacDskErr     EQU   -57     ; not a mac diskette (sig bytes are wrong)
extFSErr        EQU   -58     ; volume in question belongs to an external fs
 fsRnErr                        EQU    -59       ; file system internal error: during rename the
                                old entry was deleted but could not be restored . . .
```

```
badMDBErr       EQU   -60       ; bad master directory block
wrPermErr       EQU   -61       ; write permissions error

; Font Manager Error Codes

fontDecError    EQU   -64       ; error during font declaration
fontNotDeclared EQU   -65       ; font not declared
fontSubErr      EQU   -66       ; font substitution occured

; Disk, Serial Ports, Clock Specific Errors

firstDskErr     EQU   -84
lastDskErr      EQU   -64

noDriveErr      EQU   -64       ; drive not installed
offLinErr       EQU   -65       ; r/w requested for an off-line drive

noNybErr        EQU   -66       ; couldn't find 5 nybbles in 200 tries
noAdrMkErr      EQU   -67       ; couldn't find valid addr mark
dataVerErr      EQU   -68       ; read verify compare failed
badCkSmErr      EQU   -69       ; addr mark checksum didn't check
badBtSlpErr     EQU   -70       ; bad addr mark bit slip nibbles
noDtaMkErr      EQU   -71       ; couldn't find a data mark header
badDCkSum       EQU   -72       ; bad data mark checksum
badDBtSlp       EQU   -73       ; bad data mark bit slip nibbles
wrUnderRun      EQU   -74       ; write underrun occurred

cantStepErr     EQU   -75       ; step handshake failed
tk0BadErr       EQU   -76       ; track 0 detect doesn't change
initIWMErr      EQU   -77       ; unable to initialize IWM
twoSideErr      EQU   -78       ; tried to read 2nd side on a 1-sided drive
spdAdjErr       EQU   -79       ; unable to correctly adjust disk speed
seekErr         EQU   -80       ; track number wrong on address mark
sectNFErr       EQU   -81       ; sector number never found on a track

fmt1Err         EQU   -82       ; can't find sector 0 after track format
fmt2Err         EQU   -83       ; can't get enough sync
VerErr          EQU   -84       ; track failed to verify


clkRdErr        EQU   -85       ; unable to read same clock value twice
clkWrErr        EQU   -86       ; time written did not verify
prWrErr         EQU   -87       ; parameter ram written didn't read-verify
prInitErr       EQU   -88       ; InitUtil found the parameter ram uninitialized

rcvrErr         EQU   -89       ; SCC receiver error (framing, parity, OR)
breakRecd       EQU   -90       ; Break received (SCC)

; AppleTalk error codes

ddpSktErr       EQU   -91       ; error in soket number
```

```
ddpLenErr       EQU   -92     ; data length too big
noBridgeErr     EQU   -93     ; no network bridge for non-local send
lapProtErr      EQU   -94     ; error in attaching/detaching protocol
excessCollsns   EQU   -95     ; excessive collisions on write
portInUse       EQU   -97     ; driver Open error code (port is in use)
portNotCf       EQU   -98     ; driver Open error code (parameter RAM not configured for this
connection)
```

```
memROZErr       EQU   -99       ; hard error in ROZ

; Scrap Manager error codes

noScrapErr      EQU   -100      ; No scrap exists error
noTypeErr       EQU   -102      ; No object of that type in scrap

; Storage allocator error codes

memFullErr      EQU   -108      ; Not enough room in heap zone
nilHandleErr    EQU   -109      ; Handle was NIL in HandleZone or other;
memWZErr        EQU   -111      ; WhichZone failed (applied to free block);
memPurErr       EQU   -112      ; trying to purge a locked or non-purgeable block;

memAdrErr       EQU   -110      ; address was odd, or out of range;
memAZErr        EQU   -113      ; Address in zone check failed;
memPCErr        EQU   -114      ; Pointer Check failed;
memBCErr        EQU   -115      ; Block Check failed;
memSCErr        EQU   -116      ; Size Check failed;
memLockedErr    EQU   -117      ; trying to move a locked block (MoveHHi)

; New system error codes :

dirNFErr        EQU -120        ; Directory not found
tMWDOErr        EQU -121        ; No free WDCB available
badMovErr       EQU -122        ; Move into offspring error
wrgVolTypErr    EQU -123        ; Wrong volume type error [operation not supported for MFS]

; Resource Manager error codes (other than I/O errors)

resNotFound     EQU   -192      ; Resource not found
resFNotFound    EQU   -193      ; Resource file not found
addResFailed    EQU   -194      ; AddResource failed
addRefFailed    EQU   -195      ; AddReference failed
rmvResFailed    EQU   -196      ; RmveResource failed
rmvRefFailed    EQU   -197      ; RmveReference failed
resAttrErr      EQU   -198      ; attribute inconsistent with operation
mapReadErr      EQU   -199      ; map inconsistent with operation

;_____________________________________________________________________
;
; some miscellaneous result codes

evtNotEnb       EQU    1        ; event not enabled at PostEvent

;               System Error Alert ID definitions.   These are just for reference because
;               one cannot intercept the calls and do anything programmatically...

dsSysErr        EQU   32767     ; general system error
dsBusError      EQU   1         ; bus error
dsAddressErr    EQU   2         ; address error
```

```
dsIllInstErr    EQU  3     ; illegal instruction error
dsZeroDivErr    EQU  4     ; zero divide error
dsChkErr        EQU  5     ; check trap error
dsOvFlowErr     EQU  6     ; overflow trap error
dsPrivErr       EQU  7     ; privelege violation error
```

```
dsTraceErr      EQU   8       ; trace mode error
dsLineAErr      EQU   9       ; line 1010 trap error
dsLineFErr      EQU   10      ; line 1111 trap error
dsMiscErr       EQU   11      ; miscellaneous hardware exception error
dsCoreErr       EQU   12      ; unimplemented core routine error
dsIrqErr        EQU   13      ; uninstalled interrupt error

dsIOCoreErr     EQU   14      ; IO Core Error
dsLoadErr       EQU   15      ; Segment Loader Error
dsFPErr         EQU   16      ; Floating point error

dsNoPackErr     EQU   17      ; package 0 not present
dsNoPk1         EQU   18      ; package 1 not present
dsNoPk2         EQU   19      ; package 2 not present
dsNoPk3         EQU   20      ; package 3 not present
dsNoPk4         EQU   21      ; package 4 not present
dsNoPk5         EQU   22      ; package 5 not present
dsNoPk6         EQU   23      ; package 6 not present
dsNoPk7         EQU   24      ; package 7 not present

dsMemFullErr    EQU   25      ; out of memory!
dsBadLaunch     EQU   26      ; can't launch file

dsFSErr         EQU   27      ; file system map has been trashed
dsStknHeap      EQU   28      ; stack has moved into application heap
dsReinsert      EQU   30      ; request user to reinsert off-line volume
dsNotThe1       EQU   31      ; not the disk I wanted
negZcbFreeErr   EQU   33      ; ZcbFree has gone negative
menuPrgErr      EQU   84      ; happens when a menu is purged


;************ ADDITIONS MADE FOR NEW QUICKDRAW AND COLOR **************
; Note: the following error codes are also used but not documented anywhere obvious!!
;
; dsGreeting    EQU   40      ; welcome to Macintosh greeting
; dsFinderErr   EQU   41      ; can't load the Finder error
;

;Slot Declaration ROM Manager Errors

siInitSDTblErr  EQU   1       ;slot int dispatch table could not be initialized.
siInitVBLQsErr  EQU   2       ;VBLqueues for all slots could not be initialized.
siInitSPTblErr  EQU   3       ;slot priority table could not be initialized.

sdmJTInitErr    EQU   10      ;SDM Jump Table could not be initialized.
sdmInitErr      EQU   11      ;SDM could not be initialized.
sdmSRTInitErr   EQU   12      ;Slot Resource Table could not be initialized.
sdmPRAMInitErr  EQU   13      ;Slot PRAM could not be initialized.
sdmPriInitErr   EQU   14      ;Cards could not be initialized.
```

```
;Color Quickdraw & Color Manager Errors

cMatchErr       EQU   -150      ; Color2Index failed to find an index
cTempMemErr     EQU   -151      ; failed to allocate memory for temporary structures
cNoMemErr       EQU   -152      ; failed to allocate memory for structure
cRangeErr       EQU   -153      ; range error on colorTable request
cProtectErr     EQU   -154      ; colorTable entry protection violation
```

```
cDevErr         EQU   -155      ; invalid type of graphics device
cResErr         EQU   -156      ; invalid resolution for MakeITable

; errors for Color2Index/ITabMatch

iTabPurgErr     EQU   -9
noColMatch      EQU   -10

; errors for MakeITable

qAllocErr       EQU   -11
tblAllocErr     EQU   -12
overRun         EQU   -13
noRoomErr       EQU   -14
```