

;File TOOLEQU.TEXT

; User Interface ToolBox Equate File

; This file contains global variable and data structure
; definitions for the MacIntosh User Interface ToolBox.
; It is included with all toolBox source files and
; possibly with some application programs.

; written by Andy Hertzfeld May 5, 1982

; MODIFICATION HISTORY:

; AJH 23-May-82 Added window stuff
; AJH 22-Jun-82 Added icon stuff
; AJH 27-Jun-82 Added menu stuff
; AJH 31-Jul-82 Added control stuff; changed menu data structures
; AJH 10-Aug-82 Got rid of scrapWindow
; AJH 29-Aug-82 Added DragPattern, IShapeHook to globals, dragMsg to controls
; AJH 06-Sep-82 Added growWindow message
; AJH 12-Sep-82 Got rid of Alert and Dialog Window; added OneOne, MinusOne
; AJH 14-Sep-82 Changed for new font manager
; AJH 20-Sep-82 Added "goAway" boolean to windows; added symbolPtr globals
; BLH 24-Sep-82 Added alert, dialog globals and data structures
; AJH 05-Oct-82 Added ControlAction field to control data structure
; BLH 7-Oct-82 Added indirect globals--took main globals out of low mem
; AJH 12-Oct-82 Switched over to new font manager globals; grafport now 64 bytes
; AJH 16-Nov-82 Added TaskLock for desk manager
; AJH 21-Nov-82 LGlobals, GPortSize change for ROM 1.8
; BLH 17-Dec-82 Changed Dialog, Resource constants and globals
; BLH 19-Dec-82 Changed names for Resource upheaval
; AJH 19-Dec-82 Added CurActive, CurDeactive, deskHook globals
; AJH 24-Dec-82 Made defProcs defHandles, other data structure changes
; AJH 28-Dec-82 Broke off resource type and ID defs into separate file
; AJH 05-Jan-83 moved global base to \$E80; put dialog stuff at end
; AJH 24-Jan-83 replaced MICONLIST with MBARENABLE
; BLH 5-Feb-83 Removed alertWindow, DialogWindow globals. Changed
; Dialog data structure defs.
; AJH 5-Feb-83 added Microsoft bytes, other changes
; AJH 6-Feb-83 globals for scaling font manager
; AJH 12-Feb-83 got rid of checkFlag, made curDragAction 4 bytes long
; AJH 6-Mar-83 added GotStrike field for fontMgr (used byte from defSize)
; AJH 7-Mar-83 added scaleDisable
; AJH 5-Apr-83 added GhostWindow field in Microsoft globals
; BLH 20-Apr-83 Changed alert and dialog templates. Removed AlertKind.
; SC 20-Apr-83 Added journal stuff at end
; SC 09-May-83 Added text edit defs, changed dialogs
; AJH 10-May-83 Added CloseOrnHook, FPAddress
; AJH 25-May-83 Added AppPacks table
; AJH 31-May-83 Changed size of grafPort for new QuickDraw, Added SysResName
; AJH 02-Jun-83 Changed PicScrap to theScrap
; AJH 08-Jun-83 Added AppParmHandle
; AJH 14-Jun-83 Added DSErrCode
; SC 23-Jun-83 Removed systemKind (it's now any negative number)
; AJH 05-Jul-83 Made it match the documentation
; SC 05-Aug-83 New TE record
; AJH 05-Aug-83 Added WGIConMsg to window defProc messages
; BLH 06-Aug-83 Added ResErrProc at end of tool globals. Made LastMap
; a temporary filler.
; SC 15-Aug-83 Dropped scrapInfo, theScrap
; AJH 28-Aug-83 Changed AppleMark, CheckMark to final values

```

;      AJH 29-Aug-83   added "trackCtl" message
;      BLH  6-Sep-83   added ResReadOnly global, replacing RFiller.
;      SC   9-Sep-83   added teWdBreak for european stuff.
;      AJH 12-Sep-83   added CurDeKind
;
;-----

```

```

;
; Important Constants
;

```

```

SCREENWIDTH      .EQU    640
SCREENHEIGHT     .EQU    484
SCREENRADIUS     .EQU    $00100010    ;rounded corners for desk area
GPORTSIZE       .EQU    108          ;a grafPort is 108 bytes
;
;-----

```

```

;
; ToolBox Global Variables (kept in low memory)
;

```

```

; This section contains global variables for the font manager,
; the window manager and the menu manager
;
;-----

```

```

TOOLGBASE       .EQU    $980    ;ToolVars    ;base address of toolbox globals
LGLOBALS       .EQU    0        ;QuickDraw globals accessed 0(A5)
;
;-----

```

```

;
; Font Manager Globals
;
;-----

```

```

RomFont0       .EQU    ToolGBase    ;handle to ROM-based system font
ApFontID       .EQU    RomFont0+4   ;family ID of standard application font
GotStrike      .EQU    ApFontID+2   ;boolean indicating if we've got the strike
FMDefaultSize  .EQU    GotStrike+1  ;byte holding default size

CurFMInput    .EQU    FMDefaultSize+1 ;QuickDraw FMInput Record
CurFMFamily   .EQU    CurFMInput    ;current font family
CurFMSize     .EQU    CurFMFamily+2 ;current font size
CurFMFace     .EQU    CurFMSize+2   ;current font face
CurFMNeedBits .EQU    CurFMFace+1   ;boolean specifying whether it needs strike
CurFMDevice   .EQU    CurFMNeedBits+1 ;current font device
CurFMNumer    .EQU    CurFMDevice+2 ;current numerator of scale factor
CurFMDenom    .EQU    CurFMNumer+4  ;current denominator of scale factor

FMgrOutRec     .EQU    CurFMDenom+4  ;QuickDraw FontOutput Record
FOutError      .EQU    FMgrOutRec    ;error code
FOutFontHandle .EQU    FOutError+2   ;handle to font bits
FOutBold       .EQU    FOutFontHandle+4 ;bolding factor
FOutItalic     .EQU    FOutBold+1    ;italic factor
FOutULOffset   .EQU    FOutItalic+1  ;underline offset
FOutULShadow   .EQU    FOutULOffset+1 ;underline halo
FOutULThick    .EQU    FOutULShadow+1 ;underline thickness
FOutShadow     .EQU    FOutULThick+1 ;shadow factor
FOutExtra      .EQU    FOutShadow+1  ;extra horizontal width
FOutAscent     .EQU    FOutExtra+1   ;height above baseline
FOutDescent    .EQU    FOutAscent+1  ;height below baseline
FOutWidMax     .EQU    FOutDescent+1 ;maximum width of character
FOutLeading     .EQU    FOutWidMax+1  ;space between lines

```

```

FOutUnused .EQU FOutLeading+1 ;unused byte (must have even number)
FOutNumer .EQU FOutUnused+1 ;point for numerators of scale factor
FOutDenom .EQU FOutNumer+4 ;point for denominators of scale factor

```

```

FMDotsPerInch .EQU FOutDenom+4 ;h,v dotsPerInch of current device
FMStyleTab .EQU FMDotsPerInch+4 ;style heuristic table supplied by device
ToolScratch .EQU FMStyleTab+24 ;8 byte scratch area

```

```

;-----
;
;
; Window Manager Globals
;
;-----

```

```

WINDOWLIST .EQU TOOLSCRATCH+8 ;Z-ordered list of windows (nearest first)
SAVEUPDATE .EQU WINDOWLIST+4 ;boolean enabling update accumulation
PAINTWHITE .EQU SAVEUPDATE+2 ;boolean enabling erasing newly drawn windows
WMGRPORT .EQU PAINTWHITE+2 ;grafPort used by window manager
DESKPORT .EQU WMGRPORT+4 ;wide open port for use by applications
;
OLDSTRUCTURE .EQU DESKPORT+4 ;handle to saved structure region
OLDCONTENT .EQU OLDSTRUCTURE+4 ;handle to saved content region
GRAYRGN .EQU OLDCONTENT+4 ;handle of rounded gray desk region
SAVEVISRGN .EQU GRAYRGN+4 ;handle of temporarily saved vis region
DRAGHOOK .EQU SAVEVISRGN+4 ;proc ptr to allow user control during dragging
TEMPRECT .EQU DRAGHOOK+4 ;rectangle used as scratch by toolbox
ONEONE .EQU TempRect+8 ;long constant $00010001
MINUSONE .EQU OneOne+4 ;long constant $FFFFFFF

```

```

;-----
;
;
; General purpose bitMap for plotting bitMaps
;
;-----

```

```

ICONBITMAP .EQU MinusOne+8 ;bitmap used for plotting things

```

```

;-----
;
;
; Menu Manager Globals
;
;-----

```

```

MENULIST .EQU ICONBITMAP+14 ;pointer to current menuBar list structure
MBARENABLE .EQU MENULIST+4 ;word for enabling menuBar for deskOrns
CurDeKind .EQU MBARENABLE+2 ;kind of deactivated window
MENUFLASH .EQU CurDeKind+2 ;integer holding flash feedback count
THEMENU .EQU MENUFLASH+2 ;integer holding ID of hilited menu
SAVEDHANDLE .EQU THEMENU+2 ;handle of saved menuBits
MRMACHOOK .EQU SAVEDHANDLE+4 ;hook for Mr. MacIntosh
MENUHOOK .EQU MRMACHOOK+4 ;hook for user control during menuSelect

```

```

; Control Manager Globals

```

```

DragPattern .EQU MenuHook+4 ;pattern DragTheRgn uses
DeskPattern .EQU DragPattern+8 ;pattern to paint the desk with
DragFlag .EQU DeskPattern+8 ;implicit parameter to DragControl, 1 word
CurDragAction .EQU DragFlag+2 ;implicit actionProc parameter

```

```

; Floating Point Package Globals

```

```

FPState .EQU CurDragAction+4 ;6 bytes of floating point state

```

```

;-----
;
; Resource Manager Globals
;
;-----

```

```

TopMapHndl      .EQU    FPState+6      ;Handle to topmost map of list
SysMapHndl      .EQU    TopMapHndl+4    ;Handle to system map
SysMap          .EQU    SysMapHndl+4    ;reference number of system map
CurMap         .EQU    SysMap+2        ;reference number of current map
ResReadOnly     .EQU    CurMap+2        ;ReadOnly flag
ResLoad         .EQU    ResReadOnly+2   ;Auto-load feature
ResErr          .EQU    ResLoad+2       ;Resource error code

```

```

;-----
;
; Misc ToolBox Globals
;
;-----

```

```

TaskLock        .EQU    ResErr+2        ;flag to prevent re-entering SystemTask
FScaleDisable   .EQU    TaskLock+1      ;flag to disable font scaling
CurActivate    .EQU    FScaleDisable+1 ;window to receive activate event
CurDeactivate  .EQU    CurActivate+4   ;window to receive deactivate event
DeskHook        .EQU    CurDeactivate+4 ;hook for painting the desk

teDoText        .EQU    DeskHook+4      ;Text edit private hook
teRecal         .EQU    teDoText+4      ; ""

MicroSoft       .EQU    teRecal+4       ;12 bytes for our friends in Seattle
GhostWindow     .EQU    MicroSoft+12    ;window that FrontWindow ignores
CloseOrnHook    .EQU    GhostWindow+4   ;hook for closing desk ornaments

```

```

;-----
;
; Alert/Dialog Manager Globals
;
;-----

```

```

RestProc        .EQU    CloseOrnHook+4  ;Address of Restart failsafe procedure
SaveProc        .EQU    RestProc+4      ;Address of Save failsafe procedure
SaveSP          .EQU    SaveProc+4      ;Safe SP for restart or save
ANumber         .EQU    SaveSP+4        ;Which alert is active?
ACount          .EQU    ANumber+2       ;How many times this alert called?
DABeeper        .EQU    ACount+2        ;Beep routine
DAStrings       .EQU    DABeeper+4      ;string parameters for substitution

```

```

;-----
;
; TextEdit Globals
;
;-----

```

```

TEScrpLength    .EQU    DAStrings+16
TEScrpHandle     .EQU    TEScrpLength+4

```

```

; application package dispatch table -- 8 longWords

```

```

AppPacks      .EQU    TEsCrpHandle+4    ;handle to math code

; Misc stuff (last minute additions)

SysResName    .EQU    AppPacks+32      ;name of system.rsrc
AppParmHandle .EQU    SysResName+20    ;handle to hold application parameters
DSErrCode     .EQU    AppParmHandle+4  ;last deep shit alert ID
ResErrProc    .EQU    DSErrCode+2     ;Resource error proc.
teWdBreak     .EQU    ResErrProc+4     ;default word break routine
dlgFont       .EQU    teWdBreak+4     ;default dialog font
LastTGLocal   .EQU    dlgFont+2       ;address of last global

```

```

-----
;
;
; ToolBox Data Structure Definitions
;
; This section contains data structure definitions
; for the font manager, the window manager
; and the menu manager.
;
;
;-----

```

```

-----
;
;
; Font Manager Data Structures
;
;-----

```

```

SYSFONT      .EQU    0                ;system font is font ID 0
APPLFONT     .EQU    1                ;application font is font ID 1

```

```

; Font Manager Input Record Offsets

```

```

FMInFamily   .EQU    0                ;offset to family
FMInSize     .EQU    2                ;offset to size
FMInFace     .EQU    4                ;offset to face
FMInNeedBits .EQU    5                ;offset to needBits boolean
FMInDevice   .EQU    6                ;offset to device number
FMInNumer    .EQU    8                ;offset to numerator of scale
FMInDenom    .EQU    12               ;offset to denominator of scale

```

```

;
; Kerned Strike Font Header Data Structure
;

```

```

FFORMAT      .EQU    0                ;format of font
FMINCHAR     .EQU    2                ;lowest character in font
FMAXCHAR     .EQU    4                ;highest character in font
FMAXWD       .EQU    6                ;maximum width of characters
FBBOX        .EQU    8                ;left side of bounding box
FBBOY        .EQU    10               ;top of bounding box
FBDX         .EQU    12               ;width of bounding box
FBDY         .EQU    14               ;height of bounding box
FLENGTH      .EQU    16               ;length of font
FASCENT      .EQU    18               ;ascent above baseline
FDESCENT     .EQU    20               ;descent below baseline
FLEADING     .EQU    22               ;space between lines
FRASTER      .EQU    24               ;offset to actual bits?

```

```

;
; Definitions for Font Style Bits (in standard style byte, bits 0-6)
;

```

```

BOLDBIT      .EQU    0
ITALICBIT    .EQU    1

```

```

ULINEBIT      .EQU      2
OUTLINEBIT    .EQU      3
SHADOWBIT     .EQU      4
CondenseBit   .EQU      5
ExtendBit     .EQU      6

; Font Manager Defined control/status codes

FMgrCtl1      .EQU      8                ;printer drivers support ctl/status 8

;-----
;
; Window Manager Data Structures
;-----
;
; Window Kind Constants
;
;
;system windows are negative include desk
ornaments, etc.
DialogKind    .EQU      2                ;Dialog windows
USERKIND      .EQU      8                ;this and above numbers are for user

;
; 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
WIconMsg      .EQU      6                ;draw the grow icon

;
; Window Hit-Test Codes
;
wNoHit        .EQU      0                ;not in window at all
wInContent    .EQU      1                ;in content area of window
wInDrag       .EQU      2                ;in drag area of window
wInGrow       .EQU      3                ;in grow area of window
wInGoAway     .EQU      4                ;in go away area of window

;
; Window Data Structure Definition
;
WINDOWPORT    .EQU      0                ;GrafPort Data Structure
WINDOWKIND    .EQU      GPORTSIZE        ;integer -- logical type of window
WVISIBLE      .EQU      WINDOWKIND+2    ;boolean (1 byte) -- visible flag
WHILITED      .EQU      WVISIBLE+1      ;boolean (1 byte) -- select (hilite) flag
WGoAway       .EQU      WHILITED+1      ;boolean (1 byte) -- has go away button
WSpares       .EQU      WGoAway+1       ;spare byte reserved for future expansion
STRUCTRGN     .EQU      WSpares+1       ;handle to structure region of window
CONTRGN       .EQU      STRUCTRGN+4     ;handle to content region of window
UPDATERGN     .EQU      CONTRGN+4       ;handle to update region of window
WINDOWDEF     .EQU      UPDATERGN+4     ;handle to window definition procedure
WDATAHANDLE   .EQU      WINDOWDEF+4     ;handle to window proc-defined data
WTITLEHANDLE  .EQU      WDATAHANDLE+4   ;handle to title string
WTITLEWIDTH   .EQU      WTITLEHANDLE+4  ;width in pixels of title string (integer)
WCONROLLIST   .EQU      WTITLEWIDTH+2   ;handle to list of controls of this window
NEXTWINDOW    .EQU      WCONROLLIST+4   ;pointer to next window in z-ordered list
WINDOWPIC     .EQU      NEXTWINDOW+4    ;picture handle for updates
WREFCON       .EQU      WINDOWPIC+4     ;long integer defined by application
;

```

```
WINDOWSIZE      .EQU      WREFCON+4      ;size of window data structure
```

```
-----  
;  
; Menu Manager Data Structure Definitions  
-----
```

```
MAXMENU         .EQU      96              ;a maximum of 16*6 menus in menuBar  
MLISTSIZE       .EQU      102            ;a menu list is 102 bytes long
```

```
;  
; Menu Definition Procedure Messages  
;
```

```
MDRAWMSG        .EQU      0              ;draw yourself  
MCHOOSEMSG     .EQU      1              ;select an item  
MSizeMsg        .EQU      2              ;calculate your size
```

```
;  
; Character Definitions for MenuMarks and other special characters  
;
```

```
NOMARK          .EQU      0  
APPLEMARK       .EQU      17  
CHECKMARK       .EQU      18
```

```
;  
; MenuList Data Structure Definition -- one per menuBar  
;
```

```
LASTMENU        .EQU      0              ;number of menus in this menuList  
LASTRIGHT       .EQU      2              ;coordinate of 1st free point in menuBar  
MENUOH          .EQU      0              ;menu handle  
MENULEFT        .EQU      4              ;coordinate of left edge of menu
```

```
;  
; MenuInfoBlk Data Structure -- one per menu  
;
```

```
MENUID          .EQU      0              ;16 bit integer unique for each menuBar  
MENUWIDTH       .EQU      2              ;16 bit integer indicating menu width  
MENUHEIGHT      .EQU      4              ;16 bit integer indicating menu height  
MenuDefHandle   .EQU      6              ;handle to menu definition proc  
MENUENABLE      .EQU      10             ;longword of enable flags, one bit/item  
MENUDATA        .EQU      14             ;menu item string
```

```
;  
MENUBLKSIZE     .EQU      14             ;size of a menu block is 14 bytes plus  
;the dataString
```

```
;  
; MenuString Data Structure -- one per item  
;
```

```
ITEMICON        .EQU      0              ;offset to icon byte  
ITEMCMD         .EQU      1              ;offset to apple (command key) byte  
ITEMMARK        .EQU      2              ;offset to checkmark character byte  
ITEMSTYLE       .EQU      3              ;offset to style byte
```

```
-----  
;  
; Control Manager (button and dial) manager data structures  
-----
```

```
;  
; 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
;
; FindWindow classification 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
;
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
;

NEXTCONTROL    .EQU    0          ;handle to next control in the list
CONTRLOWNER    .EQU    NEXTCONTROL+4 ;windowPtr to owning window
CONTRLRECT     .EQU    CONTRLOWNER+4 ;bounding rectangle of control
CONTRLVIS      .EQU    CONTRLRECT+8 ;one byte boolean indicating visible state
CONTRLHILITE   .EQU    CONTRLVIS+1 ;one byte field indicating hilite state
CONTRLVALUE    .EQU    CONTRLHILITE+1 ;integer holding current value of control
CONTRLMIN      .EQU    CONTRLVALUE+2 ;integer holding minimum value of control
CONTRLMAX      .EQU    CONTRLMIN+2 ;integer holding maximum value of control
CONTRLDefHandle .EQU    CONTRLMAX+2 ;handle to control definition procedure
CONTRLDATA     .EQU    CONTRLDefHand+4 ;pointer to data handle for definition proc
ContrlAction   .EQU    ContrlData+4 ;pointer to local actionProc
CONTRLRFCON    .EQU    ContrlAction+4 ;longword refcon defined by application
CONTRLTITLE    .EQU    CONTRLRFCON+4 ;handle to title string (or string itself)

CONTRLSIZE     .EQU    CONTRLTITLE ;size of control data structure - title

```

```

;-----
;
; Alert Box Manager Data Structure Definitions
;
;-----

```

```

;Dialog record

```

```

DBounds        .EQU    0          ;Dialog bounds rectangle
DVisible       .EQU    DBounds+8 ;Visible flag
DWindProc      .EQU    DVisible+2 ;Window proc ID
DGoAway        .EQU    DWindProc+2 ;Go away flag
DRefCon        .EQU    DGoAway+2 ;Reference constant
DItems         .EQU    DRefCon+4 ;Item list ID and handle
DTitle         .EQU    DItems+2 ;Dialog window title

```

```

;Alert record

```



```

ABounds      .EQU    0           ;Alert box height and width
AItems       .EQU    ABounds+8   ;Item list ID
AStages      .EQU    AItems+2    ;Stages word

```

```

;Window subclass DialogWindow

```

```

Items        .EQU    WindowSize  ;Item list
teHandle     .EQU    Items+4     ;handle to TextEdit object
EditField    .EQU    teHandle+4  ;current field being edited
EditOpen     .EQU    EditField+2 ;Is editing open?
ADefItem     .EQU    EditOpen+2  ;default item for alerts
DWindLen     .EQU    ADefItem+2  ;Dialog window length

```

```

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

```

```

volBits      .EQU    3           ;Low 2 bits are volume (0..3)
alBit        .EQU    4           ;alert bit (put up box this time?)
OKDismissal  .EQU    8           ;bit for OK/Cancel default in each stage
                                         ;(not necessarily different for each
                                         ; stage, but saves a byte)

```

```

-----
;
; DialogList Data Structure Definitions
;
-----

```

```

DlgMaxIndex  .EQU    0           ;Maximum index (=items-1) stored here

```

```

;In each item

```

```

ItmHndl      .EQU    0           ;Handle to the item
ItmRect      .EQU    ItmHndl+4   ;Bounding rect of item
ItmType      .EQU    ItmRect+8   ;item type
ItmData      .EQU    ItmType+1   ;item string, must be even length

```

```

-----
;
; ToolBox Error Codes (passed through GrafError)
;
-----

```

```

FontDecError .EQU    -64         ;error during font declaration
FontNotDeclared .EQU    -65      ;font not declared
FontSubErr   .EQU    -66         ;font substitution occurred

```

```

-----
;
; Equates for journal interface control call
;
-----

```

```

JPlayCtl     .EQU    16         ; playBack call
JRecordCtl   .EQU    17         ; record call
JRefNum      .EQU    -2         ; refNum of journal
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)
```

```
-----
;
; Record for Text Edit object
;
-----
```

```
teDestRect   .EQU      0      ; 0 Destination rectangle (keep
together)
teViewRect   .EQU      teDestRect+8      ; 8 View rectangle rectangle

teSelRect    .EQU      teViewRect+8      ; 10 Select rectangle
teLineHite   .EQU      teSelRect+8      ; 18 Current font lineheight
teAscent     .EQU      teLineHite+2     ; 1A Current font ascent
teSelPoint   .EQU      teAscent+2      ; 1C Selection point(mouseLoc)

teSelStart   .EQU      teSelPoint+4     ; 20 selection start (KEEP together)
teSelEnd     .EQU      teSelStart+2     ; 22 selection end

teActive     .EQU      teSelEnd+2      ; 24 ST if active

teWordBreak  .EQU      teActive+2      ; 26 Word break routine
teClkProc    .EQU      teWordBreak+4    ; 2A Click loop routine

teClkTime    .EQU      teClkProc+4     ; 2E Time of last click
teClkLoc     .EQU      teClkTime+4     ; 32 location of double click

teCarTime    .EQU      teClkLoc+2      ; 33 Time for next caret toggle
teCarOn      .EQU      teCarTime+4     ; 38 Is caret on?
teCarAct     .EQU      teCarOn+1      ; 39 Is caret active?
teStyle      .EQU      teCarAct+1      ; 3A fill style

teLength     .EQU      teStyle+2      ; 3C Length of text below (Must stay
together)
teTextH      .EQU      teLength+2      ; 40 text handle goes here

teRecBack    .EQU      teTextH+4      ; 42 set true if backgnd recal
teRecLine    .EQU      teRecBack+2     ; 44 current recal line
teLftClick   .EQU      teRecLine+2     ; 46 ST if click was to left
teLftCaret   .EQU      teLftClick+1    ; 47 ST if caret was to left

teCROnly     .EQU      teLftCaret+1    ; 48 ST if CR only for line breaks

teFontStuff  .EQU      teCROnly+2     ; 50 space for font specifier
teGrafPort   .EQU      teFontStuff+8   ; 52 grafport for editing

teHiHook     .EQU      teGrafPort+4    ; 56 Hook for hilite routine
teCarHook    .EQU      teHiHook+4     ; 5A Hook for hilite routine

teNLines     .EQU      teCarHook+4     ; 5E number of lines (must keep
together)
teLines      .EQU      teNLines+2     ; 62 line starts(array of words)

TERecSize    .EQU      teLines+8      ; 6A BASE size of a record
```