```
; ============================================================
;  geoGopherCfg: configuration selector
; ============================================================
.if             Pass1
                .noeqin
                .include    geoGopherSym
                .include    geoGopherMac
                .include    geoGopher.inc
                .eqin
.endif
; ============================================================
;  constants for configuration list "dialog"
; ============================================================
CFG_TOP         =           40
CFG_BOT         =           158
CFG_LEFT        =           40
CFG_RGHT        =           264
LIST_T          =           CFG_TOP+8
LIST_B          =           CFG_BOT-8
LIST_L          =           CFG_LEFT+8
LIST_R          =           CFG_LEFT+160     ;200
MAX_LIST        =           10
; ============================================================
;  jump table
; ============================================================
                jmp         cfgStart         ;j_setup
                jmp         cfgEdit          ;j_edtcon
; ============================================================
;  Set up for configuration item scrolling.
; ============================================================
cfgStart:       LoadW       r0,connDB        ;create/open/quit
                LoadW       RecoverVector,rstrStd  ;just redraw w/50% stipple
                jsr         DoDlgBox
                lda         r0L
                cmp         #CREATE
                bne         10$
                LoadW       cfgTPtr,cfgDTxt  ;disable "connect" menu item
                LoadW       cfgDspch,cfgNop
5$              LoadW       delIcon,icnDelDs ;disable delete icon in conn. editor
                LoadW       delDspch,iconNop
                LoadB       cfgRec,$ff
                jmp         cfgEdit
10$             cmp         #OPEN
                bne         20$
                LoadW       cfgTPtr,cfgDTxt  ;disable "connect" menu item
                LoadW       cfgDspch,cfgNop
                jsr         openCfg          ;creates empty one if not found
                jsr         getCfgs
                jsr         CloseRecordFile
                lda         numItems
                beq         5$               ;empty file, jump to CREATE
                jsr         setCfg
                jsr         doCfgGui         ;loads list, sets icons & mouse
                rts
20$             cmp         #QUIT
                bne         30$              ;FIXME error handling
                jmp         doQuit
30$             rts
; ============================================================
doCreate:       lda         #CREATE          ;custom icon ID
                sta         sysDBData
                jmp         RstrFrmDialog
doExit:         lda         #QUIT            ;custom icon ID
                sta         sysDBData
                jmp         RstrFrmDialog
```

```
; ============================================================
;  Set scroll variables for this module.
; ============================================================
setCfg:         LoadB   scrlTop,LIST_T+1
                LoadB   scrlBot,LIST_B-1
                LoadW   scrlLeft,LIST_R-8
                LoadB   thmbTop,LIST_T+9
                LoadW   thmbLeft,LIST_R-6
                LoadB   thmbMax,(LIST_B-9)-(LIST_T+9)
                LoadB   maxItems,10
                rts
; ============================================================
;  Draw GUI for configuration selector and fill listbox.
; ============================================================
doCfgGui:       lda     #1              ;solid (shadow box)
                jsr     SetPattern
                LoadB   r2L,CFG_TOP+8
                LoadB   r2H,CFG_BOT+8
                LoadW   r3,CFG_LEFT+8
                LoadW   r4,CFG_RGHT+8
                jsr     Rectangle
;               ================================================
                lda     #0              ;clear
                jsr     SetPattern
                LoadB   r2L,CFG_TOP+1
                LoadB   r2H,CFG_BOT-1
                LoadW   r3,CFG_LEFT+1
                LoadW   r4,CFG_RGHT-1
                jsr     Rectangle
                LoadB   r2L,CFG_TOP
                LoadB   r2H,CFG_BOT
                LoadW   r3,CFG_LEFT
                LoadW   r4,CFG_RGHT
                lda     #$ff            ;solid line
                jsr     FrameRectangle
;               ================================================
                LoadB   r2L,LIST_T      ;listbox for filenames
                LoadB   r2H,LIST_B
                LoadW   r3,LIST_L
                LoadW   r4,LIST_R
                lda     #$ff            ;solid line
                jsr     FrameRectangle
;               ================================================
listbox:        LoadB   r3L,LIST_T
                LoadB   r3H,LIST_B
                MoveW   scrlLeft,r4
                lda     #$ff            ;solid line
                jsr     VerticalLine    ;scrollbar boundary
```

```
;		==============================================
		LoadB	r1H,LIST_T+8
		LoadW	r11,LIST_R+8
		LoadW	r0,conText
		jsr	PutString
		ldx	#0
		stx	topItem
		dex
		stx	selCfg
		LoadW	r0,cfgIcons
		jsr	DoIcons
		lda	numItems
		beq	10$
		jsr	fillList		;draws thumb
10$		php
		sei
		LoadW	otherPressVector,cfgMouse
		plp
		rts
```

```
; ==============================================================
;  Fill listbox from list of connections.
;               pass:       topItem, index of first connection to show
;               destroyed:  a0L, a1H, a2, a8, a9
; ==============================================================
fillList:       jsr         doThumb         ;trashes a8, a9
                lda         numItems        ;empty list?
                bne         10$
                lda         #0              ;clear
                jsr         SetPattern
                LoadB       r2L,LIST_T+1
                LoadB       r2H,LIST_B-1
                LoadW       r3,LIST_L+1
                lda         scrlLeft        ;r4=scrlLeft-1
                sec
                sbc         #1
                sta         r4L
                lda         scrlLeft+1
                sbc         #0
                sta         r4H
                jsr         Rectangle       ;clear listbox
                rts
10$             LoadB       a1H,LIST_T+8    ;first baseline
                LoadB       a0L,0           ;counter
                MoveB       topItem,a8
                jsr         a8Name          ;get text pointer in a8
                lda         #LIST_T+1
                sta         r2L
                sta         a2L             ;top of list item
                lda         #LIST_T+11
                sta         r2H
                sta         a2H             ;bottom of list item
20$             LoadW       r3,LIST_L+1
                lda         scrlLeft        ;r4=scrlLeft-1
                sec
                sbc         #1
                sta         r4L
                lda         scrlLeft+1
                sbc         #0
                sta         r4H
                lda         #0              ;clear
                jsr         SetPattern
                jsr         Rectangle       ;clear one list item
                CmpW        a8L,cfgEnd      ;at end of list?
                bcs         30$
                MoveW       a8,r0
                MoveB       a1H,r1H
                LoadW       r11,LIST_L+2
                jsr         PutString       ;render name in list
30$             inc         a0L
                lda         a0L
                cmp         #MAX_LIST
                bcc         40$
                rts
```

```
40$             jsr       nextName
                lda       a2L
                clc
                adc       #10           ;height of list item
                sta       a2L
                sta       r2L
                lda       a2H
                clc
                adc       #10
                sta       a2H
                sta       r2H
                AddVB     10,a1H        ;baseline
                bra       20$
```

```
; ============================================================
;  Bottom button dispatch routine.
; ============================================================
cfgBot:         lda         topItem           ;already at bottom?
                clc
                adc         #MAX_LIST
                cmp         numItems
                bcc         10$
                jsr         beep
                rts
10$             jsr         sel2Ndx
                lda         numItems
                sec
                sbc         maxItems
                sta         topItem
                jsr         fillList
                jsr         ndx2Sel           ;convert back to list index
                bmi         20$               ;not among visible items
                jsr         invList
20$             rts
; ============================================================
; Top button dispatch routine.
; ============================================================
cfgTop:         lda         topItem           ;already at top?
                bne         10$
                jsr         beep
                rts
10$             jsr         sel2Ndx
                LoadB       topItem,0
                jsr         fillList
                jsr         ndx2Sel
                bmi         20$
                jsr         invList
20$             rts
; ============================================================
;  If a visible item is selected, convert list index to item index.
; ============================================================
sel2Ndx:        lda         selCfg            ;anything visible selected?
                bmi         10$               ;nope
                clc
                adc         topItem
                sta         selIndex
10$             rts
; ============================================================
;  Convert item index back to list index and set selCfg.
;  If out of range of list, N flag will be set.
; ============================================================
ndx2Sel:        lda         selIndex
                bmi         20$
                cmp         topItem
                bcs         10$
                lda         #$ff
                bne         20$
10$             lda         selIndex
                sec
                sbc         topItem
                cmp         maxItems
                bcc         20$
                lda         #$ff
20$             sta         selCfg
                lda         selCfg            ;clear N flag from compare
                rts
```

```
; ============================================================
;  Page down routine (called through otherPressVector).
; ============================================================
cfgPgDn:        jsr         sel2Ndx
                lda         topItem
                clc
                adc         #MAX_LIST
                cmp         numItems        ;can page down at all?
                bcc         10$
                jsr         beep
                rts
10$             adc         #MAX_LIST-2
                cmp         numItems        ;can go down full page?
                bcc         20$
                lda         numItems        ;no, go to bottom
                sec
                sbc         #MAX_LIST
                sta         topItem
                bra         30$
20$             lda         topItem
                clc
                adc         #MAX_LIST-1
                sta         topItem         ;yes, top item becomes bottom
30$             jsr         fillList
                jsr         ndx2Sel
                bmi         40$
                jsr         invList         ;restore list selection display
40$             rts
; ============================================================
;  Page up routine (called through otherPressVector).
; ============================================================
cfgPgUp:        jsr         sel2Ndx
                lda         topItem         ;can page up at all?
                bne         10$
                jsr         beep
                rts
10$             cmp         #MAX_LIST-1     ;can go up full page?
                bcc         20$
                lda         topItem         ;yes,bottom item becomes top
                sec
                sbc         #MAX_LIST-1
                sta         topItem
                bra         30$
20$             lda         #0              ;no, go to top
                sta         topItem
30$             jsr         fillList
                jsr         ndx2Sel
                bmi         40$
                jsr         invList         ;restore list selection display
40$             rts
```

```
; ============================================================
;  Check for mouse click over listbox/scroll bar.
;               destroyed: a0
; ============================================================
cfgMouse:       lda     mouseData
                bpl     20$
10$             rts                     ;ignore releases
20$             lda     numItems        ;empty list?
                beq     10$
                LoadB   r2L,LIST_T+1
                LoadB   r2H,LIST_T+11
                LoadW   r3,LIST_L+1
                lda     scrlLeft        ;r4=scrlLeft-1
                sec
                sbc     #1
                sta     r4L
                lda     scrlLeft+1
                sbc     #0
                sta     r4H
                ldy     #0
                php
                sei
30$             jsr     IsMseInRegion
                cmp     #$ff
                beq     50$             ;hit
                iny
                cpy     #MAX_LIST
                bne     40$
                plp                     ;click wasn't over listbox
                bra     ckScroll
40$             AddVB   10,r2L          ;height of list item
                AddVB   10,r2H
                bra     30$
;               ================================================
;               click over config name: invert, check for double-click
;               ================================================
50$             LoadB   dblClickCount,#CLICK_COUNT
                plp
                tya                     ;list index
                clc
                adc     topItem
                cmp     numItems        ;click over empty area?
                bcs     90$             ;yes, exit
                cpy     selCfg          ;already selected?
                beq     70$             ;yes, check for double-click
                tya
                ldx     selCfg          ;anything selected?
                bmi     60$             ;no
                pha                     ;save list index
                txa
                jsr     invList         ;deselect previous
                pla
60$             sta     selCfg
                jsr     invList         ;select new one
70$             lda     dblClickCount   ;check for double-click
                beq     90$             ;double-click timed out?
                lda     mouseData
                bpl     70$             ;still pressed?
80$             lda     dblClickCount   ;no, released
                beq     90$             ;double-click timed out?
                lda     mouseData       ;no, check for 2nd click
                bmi     80$             ;still released?
                jmp     readCfg         ;double-click, read the record
90$             rts
ckScroll:       LoadB   r2L,LIST_T+8    ;click above scroll thumb?
                ldx     thumbSav
                dex
                stx     r2H
                lda     scrlLeft        ;r3=scrlLeft+1
                clc
                adc     #1
                sta     r3L
                lda     scrlLeft+1
                adc     #0
                sta     r3H
                LoadW   r4,LIST_R+1
                php
                sei
                jsr     IsMseInRegion
                plp
                cmp     #$ff
                bne     10$
                jsr     cfgPgUp
                rts
10$             lda     thumbSav        ;click below scroll thumb?
                clc
                adc     thumbHi
                sta     r2L
                LoadB   r2H,LIST_B-8
                lda     scrlLeft        ;r3=scrlLeft+1
                clc
                adc     #1
                sta     r3L
                lda     scrlLeft+1
                adc     #0
                sta     r3H
                LoadW   r4,LIST_R-1
                php
                sei
                jsr     IsMseInRegion
                plp
                cmp     #$ff
                bne     20$
                jsr     cfgPgDn
20$             rts
; ============================================================
;  Invert listbox selection.
;               pass:   .A, listbox index of name to invert
; ============================================================
invList:        sta     a8L             ;dest
                LoadB   a9L,10          ;src (name height in pixels)
                ldx     #a8L
                ldy     #a9L
                jsr     BBMult
                lda     #LIST_T+1
                clc
                adc     a8L
                sta     r2L
                lda     #LIST_T+11
                clc
                adc     a8L
                sta     r2H
                LoadW   r3,LIST_L+1
                lda     scrlLeft        ;r4=scrlLeft-1
                sec
                sbc     #1
                sta     r4L
                lda     scrlLeft+1
                sbc     #0
                sta     r4H
                jsr     InvertRectangle
                rts
```

```
; ============================================================
;  Clear selector "dialog" area.
; ============================================================
clrCfg:         lda         #2                ;50% stipple
                jsr         SetPattern
                LoadB       r2L,CFG_TOP
                LoadB       r2H,CFG_BOT+8 ;shadow box
                LoadW       r3,CFG_LEFT
                LoadW       r4,CFG_RGHT+8
                jsr         Rectangle
                rts
; ============================================================
;  Handler for "Open" icon in selector.
; ============================================================
readCfg:        jsr         getCfg            ;from selCfg
                bcc         20$
                jsr         doCfgGui          ;redraw config selector
                rts
20$             ldx         #0
                ldy         #0
30$             lda         cfgBuf,x          ;go past name
                beq         40$
                inx
                bne         30$
40$             inx
50$             lda         cfgBuf,x
                sta         hostname,y
                beq         60$
                inx
                iny
                bne         50$
60$             inx
                ldy         #0
70$             lda         cfgBuf,x
                sta         port,y
                beq         80$
                inx
                iny
                bne         70$
80$             jsr         clrCfg
                jsr         clrStat
                jsr         setGoph
                jmp         connect
; ============================================================
;  Handler for "Edit" icon in selector.
; ============================================================
editCfg:        jsr         getCfg            ;from selCfg
                bcc         10$
                jsr         doCfgGui          ;redraw config selector
                rts
10$             jsr         clrCfg
                jsr         bufToCfg
                LoadW       delIcon,icnDel    ;enable delete icon
                LoadW       delDspch,doDelete
                jmp         cfgEdit           ;to configuration editor
; ============================================================
;  Handler for "Cancel" icon in selector.
; ============================================================
cnclCfg:        LoadW       cfgTPtr,cfgTxt    ;restore "connect" menu item
                LoadW       cfgDspch,doCfg
                jsr         clrCfg
                jsr         clrStat
                jmp         cfgStart
```

```
; =============================================================
;  Read configuration names and record numbers into a table.
;               pass:     call openCfg first
;               return:   cfgNames, numItems populated
;               destroyed: a8
; =============================================================
getCfgs:        LoadB     numItems,0
                LoadW     a8,cfgNames
                ldx       #4                ;first record is driver no.
10$             lda       fileHeader,x      ;VLIR index
                beq       50$               ;record in use?
                txa
                clc
                ror       a
                tay
                dey
                sty       curRecord
                LoadW     r7,cfgBuf
                LoadW     r2,69             ;FIXME use constants
                txa
                pha
                jsr       ReadRecord
                txa
                beq       20$
                pla
                LoadW     a8,cfgsErr
                txa                         ;restore
                jsr       showCode
                jmp       EnterDeskTop
20$             pla
                tax
                ldy       #0
30$             lda       cfgBuf,y
                sta       (a8),y
                beq       40$
                iny
                bne       30$
40$             iny
                lda       curRecord
                sta       (a8),y
                iny
                tya
                clc
                adc       a8L
                sta       a8L
                lda       #0
                adc       a8H
                sta       a8H
                inc       numItems
50$             inx
                inx
                bne       10$
                MoveW     a8,cfgEnd
                rts
```

```
; ============================================================
;  Read selected configuration record.
;               pass:       list index at selCfg
;               return:     selected record at cfgBuf
; ============================================================
getCfg:         lda         selCfg
                cmp         #$ff
                bne         10$
                LoadW       errMsg,noSel
                LoadW       r0,errorDB
                LoadW       RecoverVector,rstrCon  ; redraw behind editor
                jsr         DoDlgBox
                sec
                rts
10$             clc
                adc         topItem
                sta         a8L
                jsr         a8Name
                jsr         a8Rec           ;get VLIR record no.
                sta         cfgRec
                jsr         openCfg
                MoveB       cfgRec,curRecord
                LoadW       r7,cfgBuf
                LoadW       r2,72           ;FIXME use constant
                jsr         ReadRecord
                txa
                beq         20$
                brk                         ;FIXME error handling
20$             jsr         CloseRecordFile
                clc
                rts
; ============================================================
;  Get pointer to config name at given index.
;               pass:       a8L, index into config name list
;               return:     a8, pointer to filename
; ============================================================
a8Name:         ldx         a8L
                LoadW       a8,cfgNames
                txa
                bne         10$
                rts                         ;return address of 0th one
10$             ldy         #0
20$             lda         (a8),y
                beq         30$
                iny
                bne         20$
30$             iny                         ;past null-terminator
                iny                         ;past record no.
                tya
                clc
                adc         a8L
                sta         a8L
                lda         #0
                adc         a8H
                sta         a8H
                dex
                bne         10$
                rts
```

```
; ==============================================================
;  Get record number for configuration name.
;               pass:       a8, address of configuration name in table.
;               return:     .A, record number
; ==============================================================
a8Rec:          ldy         #0
10$             lda         (a8),y
                beq         20$
                iny
                bne         10$
20$             iny
                lda         (a8),y
                rts
; ==============================================================
;  Get next configuration name from table.
;               pass:       a8, address of configuration name in table
;               return:     a8, address of next configuration name
; ==============================================================
nextName:       ldy         #0
10$             lda         (a8),y
                beq         20$
                iny
                bne         10$
20$             iny                           ;past null terminator
                iny                           ;past record no.
                tya
                clc
                adc         a8L
                sta         a8L
                lda         #0
                adc         a8H
                sta         a8H
                rts
; ==============================================================
;  Kill text input.
; ==============================================================
killText:       php
                sei
                LoadW       otherPressVector,0
                plp
                LoadB       getting,0         ;flag to ignore carriage return
                LoadB       keyData,#$0d
                lda         keyVector
                ldx         keyVector+1
                jsr         CallRoutine       ;simulate hitting Enter
                LoadW       rightMargin,319
                rts
; ==============================================================
;  Restore screen under standard dialog (or configuration selector).
; ==============================================================
rstrStd:        LoadB       r2L,DEF_DB_TOP
                LoadB       r2H,DEF_DB_BOT+8
                LoadW       r3,DEF_DB_LEFT
                LoadW       r4,DEF_DB_RIGHT+8
                bra         rstr
rstrCon:        LoadB       r2L,CFG_TOP
                LoadB       r2H,CFG_BOT+8
                LoadW       r3,CFG_LEFT
                LoadW       r4,CFG_RGHT+8
rstr:           lda         #2                ;50% stipple
                jsr         SetPattern
                jsr         Rectangle
                LoadW       RecoverVector,stdDone  ;don’t repeat for shadow
stdDone:        rts
```

```
noSel:          .byte     "No connection selected.",0
; ============================================================
;  icons for configuration selector
; ============================================================
cfgIcons:       .byte     5               ;no. icons
                .word     0               ;don’t reposition mouse
                .byte     0
;               ================================================
                .word     topArrow        ;icon image in resident module
                .byte     (LIST_R/8)-1,LIST_T ;X (cards), Y (pixels)
                .byte     1,8             ;width (cards), height (pixels)
                .word     cfgTop          ;dispatch routine
;               ================================================
                .word     botArrow        ;icon image in resident module
                .byte     (LIST_R/8)-1,LIST_B-7 ;X (cards), Y (pixels)
                .byte     1,8             ;width (cards), height (pixels)
                .word     cfgBot          ;dispatch routine
;               ================================================
                .word     icnOpen         ;pointer to icon image
                .byte     (LIST_R+8)/8,LIST_T+24  ;X (cards), Y (pixels)
                .byte     6,16            ;width (cards), height (pixels)
                .word     readCfg         ;dispatch routine
;               ================================================
                .word     icnEdit         ;pointer to icon image
                .byte     (LIST_R+8)/8,LIST_T+48  ;X (cards), Y (pixels)
                .byte     6,16            ;width (cards), height (pixels)
                .word     editCfg         ;dispatch routine
;               ================================================
                .word     icnCncl         ;pointer to icon (use editor’s)
                .byte     (LIST_R+8)/8,LIST_T+72  ;X (cards), Y (pixels)
                .byte     6,16            ;width (cards), height (pixels)
                .word     cnclCfg         ;dispatch routine
;               ================================================
icnOpen:
                          Open
icnEdit:
                          Edit

;               ================================================
conText:        .byte     "Select",GOTOXY
                .word     LIST_R+8
                .byte     LIST_T+18,"connection:",0
```