

```

; =====
; geoGopherRes: resident module
; =====
.if          Pass1
        .include    geoGopherSym
        .include    geoGopherMac
        .include    geoGopher.inc
        .include    ultimate.inc
.endif
; =====
        jmp         start
        nop
; =====
pageDown: .block    2
pageUp:   .block    2
; =====
start:   LoadB     dispBufferOn,#ST_WR_FORE
        MoveB      curDrive,drvSave
        jsr        getMods          ;get module T&S
        MoveB      permName+13,infoMsg1+10 ;copy version number
        MoveB      permName+15,infoMsg1+12 ;to info dialog
        jsr        setGoph          ;set up for gopher item scroll
        lda        #2                ;50% stipple
        jsr        SetPattern
        LoadB      r2L,0
        LoadB      r2H,199
        LoadW      r3,0
        lda        scrLLeft          ;r4 = scrLLeft-1
        sec
        sbc        #1
        sta        r4L
        lda        scrLLeft+1
        sbc        #0
        sta        r4H
        jsr        Rectangle        ;clear screen
        jsr        titleBar
        LoadW      r0,mainMenu
        php
        sei
        PushW      mouseXPos
        PushB      mouseYPos
        lda        #0
        jsr        DoMenu
        PopB       mouseYPos
        PopW       mouseXPos
        plp
        jsr        drawStat
        jsr        drawScrl
        LoadW      r0,itmlcons
        LoadW      topDspch,0       ;disable top/bottom icons
        LoadW      botDspch,0       ;(dummy scroll arrows only)
        jsr        Dolcons

```

```

        jsr      ckVersion
        bcc      10$
        jmp      EnterDeskTop
10$      LoadW    r7,modLoad
        lda      #MOD_ULT
        jsr      swapMod      ;exits on fail
        jsr      j_probe
        bcc      30$
        LoadW    r0,noUltDlg
        jsr      DoDlgBox      ;don't need RecoverVector here
        jmp      EnterDeskTop
30$      ldx      #0      ;initialize text Y pos. table
        lda      #ITEM_Y_1
40$      sta      icnYPsns,x
        clc
        adc      #ITEM_HI
        inx
        cpx      maxItems
        bne      40$
        rts      ;to MainLoop

; =====
; DYN_SUB_MENU handlers
; =====
geosDyn:  jsr      saveMenu
        LoadW    RecoverVector,rstrMenu
        LoadW    r0,geosMenu
        rts
optDyn:   jsr      saveMenu
        LoadW    RecoverVector, rstrMenu
        LoadW    r0,optMenu
        rts

; =====
; "info" menu handler
; =====
doInfo:   jsr      GotoFirstMenu
        LoadW    r0,infoDB
        LoadW    RecoverVector,rstrDlg
        jsr      DoDlgBox
        rts

; =====
; "network" menu handler
; =====
doNet:    jsr      GotoFirstMenu
        LoadW    r0,netDB
        LoadW    RecoverVector,rstrDlg
        jsr      DoDlgBox
        rts

; =====
; "connect" menu handler (show configuration dialogs)
; Configuration dialogs will either exit program or call "connect".
; =====
doCfg:    jsr      GotoFirstMenu
        LoadW    r7,modLoad      ;address to load setup module
        lda      #MOD_CFG
        jsr      swapMod
        LoadW    topDspch,0      ;disable top/bottom icons
        LoadW    botDspch,0
        jmp      j_setup      ;will only return via jump
cfgNop:   jsr      GotoFirstMenu
        rts      ;for disabled menu item

```

```

; =====
; "close" menu handler
; =====
doClose:      jsr      GotoFirstMenu
               LoadW   r0,closeMsg
               jsr      showStat
               LoadB   numItems,0
               jsr      clrItems
               jsr      drawScrl
               php
               sei
               LoadW   topDspch,0      ;disable top/bottom icons
               LoadW   botDspch,0
               jsr      disHome        ;disable HOME icon
               jsr      disBack        ;disable BACK icon
               LoadW   otherPressVector,0 ;disable scroll paging
               LoadW   cfgTPtr,cfgTxt  ;restore main menu
               LoadW   cfgDspch,doCfg
               plp
               jsr      clrStat
               lda      #0
               sta      hostname
               jsr      titleBar
               rts

; =====
; "quit" menu handler
; =====
doQuit:        jsr      GotoFirstMenu
               lda      drvSave        ;drive program loaded from
               cmp      curDrive
               beq      10$
               jsr      SetDevice
               jsr      OpenDisk
10$:           jmp      EnterDeskTop

```

```

; =====
; Connect to selected gopher server. If the config dialog returns
; at all, it will be via jump to here. The only other execution path
; is a jump to doQuit to exit the program.
; =====
connect:    lda        drvSave        ;drive program loaded from
            cmp        curDrive
            beq        20$
            jsr        SetDevice
            txa
            bne        10$
            jsr        OpenDisk
            txa
            beq        20$
10$         pha
            LoadW      a8,drvErr
            pla
            jsr        showCode
            jmp        EnterDeskTop
20$         LoadW      r7,modLoad      ;address to load network driver
            lda        #MOD_ULT
            jsr        swapMod        ;FIXME error handling
            LoadW      r0,openMsg
            jsr        showStat
            jsr        j_conn        ;in VLIR jump table
            bcc        25$
23$         LoadW      errMsg,statBuf
            LoadW      RecoverVector,rstrDlg
            LoadW      r0,errorDB
            jsr        DoDlgBox
            jmp        doClose
25$         LoadW      a7,crLfOnly    ;send CR/LF for gophermap
            ldx        #2
            lda        socket
            jsr        j_write
            bcs        23$
            LoadW      cfgTPtr,closeTxt ;change "connect" menu item to "close"
            LoadW      cfgDspch,doClose
            jsr        getItems
            bcs        30$
            jsr        cntlItems
            bcc        40$
            cmp        #BUF_END
            beq        40$
30$         LoadW      errMsg,statBuf
            LoadW      r0,errorDB
            LoadW      RecoverVector,rstrDlg
            jsr        DoDlgBox
            jmp        doClose
40$         jsr        titleBar
            ldx        #0
            jsr        dolItems
            jsr        j_close
            jsr        clrStat
            jsr        pushHome
            jsr        enHome        ;serves as "refresh" here
            rts

; =====
; handler for HOME icon
; =====
doHome:     jsr        disBack
            jmp        connect

```

```

=====
; Make home entry at start of "back stack".
=====
pushHome:  ldx      #0
           ldy      #0
10$         lda      hostname,x
           sta      backStak,y
           beq      20$
           inx
           iny
           bne      10$
20$         iny
           lda      #TYP_FLDR
           sta      backStak,y
           iny
           lda      #0
           sta      backStak,y      ;null selector
           iny
           sta      backStak,y      ;placeholder for topltem
           iny
           ldx      #0
30$         lda      port,x
           sta      backStak,y
           beq      40$
           inx
           iny
           bne      30$
40$         LoadB   backNdx,1
           jsr      disBack
           rts

```

```

; =====
; Icon handler for gopher item selection.
; =====
doSelect:    lda        r0L            ;icon no. selected
             clc
             adc        topltem
             tax
             jsr        getItem        ;to a0
             bcc        10$
             rts                ;no items available
10$          ldy        #0
             lda        (a0),y
             sta        itemType
             cmp        #TYP_INFO
             bne        20$
             inc        a0L            ;past type char.
             bne        15$
             inc        a0H
15$          jsr        saveItms        ;save item state
             LoadB      textType,TXT_INFO
             LoadB      tooBig,0        ;assume there won't be that many!
             LoadW      r0,vwTextDB
             LoadW      RecoverVector,rstrTDlg
             jsr        DoDlgBox        ;calls showText via DB_USR_ROUT
             rts
20$          iny
30$          lda        (a0),y        ;past display string
             cmp        #9            ;tab
             beq        40$
             iny
             bne        30$
40$          iny                ;to selector
             tya
             clc
             adc        a0L
             sta        a0L
             lda        a0H
             adc        #0
             sta        a0H            ;a0 now holds selector
             jsr        chkType        ;gopher type supported?
             bcc        tySelect
             LoadW      r0,errorDB
             LoadW      RecoverVector,rstrDlg
             jsr        DoDlgBox
             rts

```

```

tySelect:  lda      popping      ;are we popping from the stack?
          bne      10$          ;if so, a1 already holds hostname
          jsr      a1Host      ;hostname in selector to a1
10$       jsr      newHost      ;check if hostname changed
          bcc      20$
          rts
20$       lda      itemType
          cmp      #TYP_SRCH
          bne      50$
          LoadB    srchText,0
          LoadW    a9,srchText
          LoadW    r0,searchDB
          LoadW    RecoverVector,rstrDone ;screen will be redrawn
          jsr      DoDlgBox      ;prompt for search text
          lda      r0L
          cmp      #CANCEL
          bne      30$
          rts
30$       lda      srchText
          bne      40$
          LoadW    errMsg,noSrch
          LoadW    r0,errorDB
          LoadW    RecoverVector,rstrDlg
          jsr      DoDlgBox
          rts
40$       jsr      makeSSel      ;create search selector
          ;
          =====
50$       LoadW    r0,readMsg
          jsr      showStat
          jsr      j_conn
          bcc      60$
          jsr      j_close
          rts
60$       lda      popping
          bne      80$          ;selector formatted, don't re-push
          lda      itemType
          cmp      #TYP_SRCH
          beq      70$          ;selector already formatted
          MoveW    a0,r0
          jsr      saveChar      ;null-terminate selector
70$       lda      itemType
          cmp      #TYP_FLDR      ;only folders change our "location"
          bne      80$          ;and therefore require a push
          jsr      pushBack
          ;
          =====
80$       ldy      #0
90$       lda      (a0),y
          beq      100$
          sta      dataBuf,y
          iny
          bne      90$
100$      lda      #$0d
          sta      dataBuf,y
          iny
          lda      #$0a
          sta      dataBuf,y
          iny
          tya
          tax          ;length to send
          LoadW    a7,dataBuf
          lda      socket
          jsr      j_write      ;send selector
          php
          lda      itemType
          cmp      #TYP_SRCH
          beq      110$          ;didn't saveChar
          lda      popping

```

```

    bne      110$
    MoveW    a0,r0
    jsr      restChar      ;restore selector terminator
110$    plp
    bcc      doType
    rts      ;FIXME need dialog here?
;
a1Host: ldy      #0      ;hostname in selector to a1
10$     lda      (a0),y
    cmp      #9      ;find end of selector (tab)
    beq      30$
    iny
    bne      10$
30$     iny      ;to hostname
    tya
    clc
    adc      a0L
    sta      a1L
    lda      a0H
    adc      #0
    sta      a1H
    rts
; =====
doType: lda      itemType
    cmp      #TYP_FLDR
    beq      10$
    cmp      #TYP_SRCH
    bne      40$
;
10$     jsr      getItems
    bcs      20$
    jsr      cntItems
    bcc      30$
    cmp      #BUF_END
    beq      30$
    cmp      #BUF_EMPTY
    bne      20$
    LoadW    errMsg,noData
    LoadW    r0,errorDB
    LoadW    RecoverVector,rstrDlg
    jsr      DoDlgBox
    jsr      clrStat
    rts
20$     LoadB    popping,0      ;if we were, we're not any more
    LoadW    errMsg,badItems
    LoadW    r0,errorDB
    LoadW    RecoverVector,rstrDlg
    jsr      DoDlgBox
    lda      backNdx      ;must restore variables
    bne      25$
    jmp      doClose
25$     jsr      popBack      ;restore previous location
    rts
30$     lda      popping
    beq      35$
    LoadB    popping,0
    lda      topBack
35$     tax
    jsr      doltems
    jsr      j_close
    jsr      clrStat
    rts
;
40$     cmp      #TYP_BIN
    beq      50$
    cmp      #TYP_TEXT
    bne      120$      ;FIXME error handling

```


50\$	jsr	readCvt	
	bcc	100\$	
	cmp	#CNCL_ERR	;cancelled (not an error)
	beq	60\$	
	cmp	#CLOSE	;closed text file viewer
	bne	70\$	
60\$	jsr	clrStat	
	rts		
70\$	jsr	byte2asc	;error code in .A
	ldx	#0	
80\$	lda	ascNum,x	
	sta	errCode,x	
	beq	90\$	
	inx		
	bne	80\$	
90\$	LoadW	errMsg,badCode	;FIXME provide error messages
	LoadW	r0,errorDB	
	LoadW	RecoverVector,rstrDlg	
	jsr	DoDlgBox	
	bra	110\$	
100\$	LoadW	okMsg,dIOK	
	LoadW	r0,okDB	
	LoadW	RecoverVector,rstrDlg	
	jsr	DoDlgBox	
110\$	jsr	clrStat	
120\$	rts		

```

=====
; Check if a new host is specified. If so, set hostname and port.
;
;      pass:      a1, address of host in selector or stack
;      return:    carry set on error, clear otherwise
;      destroyed: a1
=====
newHost:  lda      itemType
          cmp      #TYP_INFO
          bne      10$
          clc
          rts
10$      idx      #0
          ldy      #0
20$      lda      hostname,x      ;new hostname?
          bne      40$
          lda      (a1),y
          beq      30$             ;in stack
          cmp      #9             ;in selector
          bne      50$             ;different length
30$      clc
          rts                     ;same hostname
40$      cmp      (a1),y
          bne      50$
          inx
          iny
          bne      20$
50$      idx      #0              ;hostname has changed
          ldy      #0
60$      lda      (a1),y
          beq      70$             ;if popping from stack
          cmp      #9
          beq      70$
          sta      hostname,x
          inx
          iny
          cpy      #MAX_HOST
          bcc      60$
          LoadW    errMsg,longHost
          LoadW    r0,errorDB
          LoadW    RecoverVector,rstrDlg
          jsr      DoDlgBox
          sec
          rts
70$      lda      #0
          sta      hostname,x
          lda      popping
          beq      90$
          iny              ;to type (back stack)
          iny              ;to topltem
          iny              ;to selector
80$      lda      (a1),y          ;past selector to port
          beq      90$
          iny
          bne      80$

```

90\$	iny		
	ldx	#0	
100\$	lda	(a1),y	;copy port number
	beq	110\$;if popping from stack
	cmp	#\$0d	
	beq	110\$	
	cmp	#9	;Gopher+ (tab, +, CR/LF)
	beq	110\$	
	sta	port,x	
	inx		
	iny		
	bne	100\$	
110\$	lda	#0	
	sta	port,x	
	jsr	titleBar	;update hostname in title bar
	clc		
	rts		

```

; =====
; Create search selector by appending tab and search text.
;
;      pass:      a0, selector portion of current gopher item
;      return:    a0, constructed search selector
; =====
makeSSel:    ldy      #0
10$          lda      (a0),y
             sta      srchSel,y
             cmp      #9
             beq      20$
             iny
             bne      10$
20$          iny
             ldx      #0
30$          lda      srchText,x
             sta      srchSel,y
             beq      40$
             inx
             iny
             bne      30$
40$          LoadW    a0,srchSel
             rts

; =====
; Check for supported gopher types.
; =====
chkType:     lda      itemType      ;type requested
             cmp      #TYP_TEXT
             bne      10$
             clc
             rts
10$          cmp      #TYP_FLDR
             bne      20$
             clc
             rts
20$          cmp      #TYP_ERR
             bne      30$
             LoadW    errMsg,errItem
             sec
             rts
30$          cmp      #TYP_SRCH
             bne      40$
             clc
             rts
40$          cmp      #TYP_BIN
             bne      50$
             clc
             rts
50$          cmp      #TYP_INFO
             bne      60$
             clc
             rts
60$          LoadW    errMsg,noType ;fall-through: unknown type
             sec
             rts

```

```

; =====
; Clear items area (between menu and status area).
; =====
clrItems:    lda        #2                ;50% stipple
            jsr        SetPattern
            MoveB      scr1Top,r2L
            MoveB      scr1Bot,r2H
            LoadW      r3,0
            lda        scr1Left          ;r4 = scr1Left-1
            sec
            sbc        #1
            sta        r4L
            lda        scr1Left+1
            sbc        #0
            sta        r4H
            jsr        Rectangle
            rts

; =====
; Enable and draw BACK icon.
; =====
enBack:      LoadW      backIcon,back
drawBack:    LoadB      r1L,35          ;X (cards)
            LoadB      r1H,189         ;Y (pixels)
            LoadB      r2L,4           ;width (cards)
            LoadB      r2H,9           ;height (pixels)
            PushW      r0              ;don't trash showStat text
            LoadW      r0,back
            jsr        BitmapUp
            PopW       r0
            rts

; =====
; Disable and hide BACK icon.
; =====
disBack:     LoadW      backIcon,0
            lda        #0              ;clear
            jsr        SetPattern
            LoadB      r2L,189
            LoadB      r2H,197
            LoadW      r3,280
            LoadW      r4,311
            jsr        Rectangle
            rts

; =====
; Enable and draw HOME icon.
; =====
enHome:      LoadW      homeIcon,home
drawHome:    LoadB      r1L,30         ;X (cards)
            LoadB      r1H,189         ;Y (pixels)
            LoadB      r2L,4           ;width (cards)
            LoadB      r2H,9           ;height (pixels)
            PushW      r0              ;don't trash showStat text
            LoadW      r0,home
            jsr        BitmapUp
            PopW       r0
            rts

; =====
; Disable and hide HOME icon.
; =====
disHome:     LoadW      homeIcon,0
            lda        #0              ;clear
            jsr        SetPattern
            LoadB      r2L,189
            LoadB      r2H,197
            LoadW      r3,240
            LoadW      r4,271
            jsr        Rectangle
            rts

```

```

; =====
; Draw and/or clear status area at bottom of screen.
; =====
clrStat:    clc
            bcc     clrStat2
drawStat:   sec
clrStat2:   php
            lda     #0             ;clear
            jsr     SetPattern
            LoadB   r2L,STATUS_T+1
            LoadB   r2H,198
            LoadW   r3,0
            LoadW   r4,319
            jsr     Rectangle
            lda     homelcon        ;was HOME icon enabled?
            ora     homelcon+1
            beq     10$
            jsr     drawHome
10$:        lda     backlcon        ;was BACK icon enabled?
            ora     backlcon+1
            beq     20$
            jsr     drawBack
20$:        plp
            bcs     30$
            rts
30$:        LoadW  r3,0
            LoadW  r4,319
            lda     #STATUS_T
            sta     r11L
            sta     r11H
            lda     #0
            sec
            jsr     DrawLine
            LoadW  r3,0
            LoadW  r4,319
            lda     #199
            sta     r11L
            sta     r11H
            lda     #0
            sec
            jsr     DrawLine
            rts
; =====
; Show status message at bottom of screen.
;
;         pass:    r0, address of message
;         return:   nothing
; =====
showStat:   jsr     clrStat
            LoadB  r1H,STATUS_T+8
            LoadW  r11,3
            jsr     PutString
            rts

```

```

; =====
; Draw title bar at top of screen.
; =====
titleBar:    PushW    a0          ;strWidth trashes it
             lda      permName+13 ;version number
             sta      title+11
             lda      permName+15
             sta      title+13
             lda      #0
             sta      title+15      ;remove hostname if present
             lda      #9          ;horizontal stripes
             jsr      SetPattern
             LoadB    r2L,0
             LoadB    r2H,14
             lda      mainMenu+4    ;width of main menu
             clc
             adc      #1
             sta      r3L
             lda      mainMenu+5
             adc      #0
             sta      r3H
             LoadW    r4,319
             jsr      Rectangle
             lda      hostname      ;copied from input dialog
             beq      30$
             lda      #'['
             sta      title+15
             ldx      #0
10$          lda      hostname,x
             beq      20$
             sta      server,x      ;in title string
             inx
20$          bne      10$
             lda      #']'
             sta      server,x
             lda      #' '
             sta      server+1,x
             lda      #0
             sta      server+2,x
30$          LoadW    r0,title
             jsr      strWidth      ;returns string width in a0
             LoadW    r1H,9        ;baseline
             LoadW    r11,320
             SubW      mainMenu+4,r11 ;320 minus menu width
             SubW      a0,r11      ;minus string width
             clc
             ror      r11H
             ror      r11L          ;divided by two
             AddW      mainMenu+4,r11 ;plus menu width (i.e. right edge)
             PushW     r11
             MoveW     r11,r3      ;clear text area first
             LoadB    r2L,1
             LoadB    r2H,13
             lda      r3L
             clc
             adc      a0L
             sta      r4L
             lda      r3H
             adc      a0H
             sta      r4H
             lda      #0          ;clear
             jsr      SetPattern
             jsr      Rectangle
             PopW      r11
             jsr      PutString     ;now render the string
             PopW      a0          ;strWidth trashed it
             rts

```

```

=====
; Check if bytes pointed to by a0 are a gopher item.
;
;      pass:      a0, address to test (max. 255 characters)
;      return:     carry set if valid item, clear otherwise
;                  if valid, item type in .A, .X points past end
;
=====
isltem:      ldx      #0
            ldy      #0
            lda      (a0),y      ;item type
10$          cmp      itmTypes,x
            beq      30$
            inx
            cpx      #NUM_TYPE
            bne      10$
20$          clc
            rts
30$          ldx      #3          ;three field delimiters
40$          iny
            beq      20$
            lda      (a0),y
            cmp      #9          ;tab
            bne      40$
            dex          ;next field
            bne      40$
50$          iny
            lda      (a0),y      ;verify numeric port
            cmp      #'0'
            bcc      60$
            cmp      #'9'+1
            bcs      20$
            bcc      50$
60$          cmp      #9          ;Gopher+ (tab, +, CR/LF)
            bne      70$
            iny          ;plus
            iny          ;CR/LF
70$          cmp      #$0d        ;trailing CR/LF
            bne      20$
            iny
            lda      (a0),y
            cmp      #$0a
            bne      20$
            iny
            tya
            tax
            ldy      #0
            lda      (a0),y
            sec
            rts

```