```
'Make a custom control hyperlink with a Liberty BASIC graphicbox.
'-----------------------------------------------------------------'
 [SetupWindow]
 NoMainWin
 WindowWidth = 400
 WindowHeight = 200
 UpperLeftX = Int((DisplayWidth-WindowWidth)/2)
 UpperLeftY = Int((DisplayHeight-WindowHeight)/2)
 Link1.Width = 80
 Link1.Height = 26
 GraphicBox #Win.Link1, 4, 4, Link.Width, Link.Height
 Stylebits #Win.Link1, 0, _WS_BORDER, 0, 0
 Open "Hyperlink Custom Control" For Window As #Win
 #Win, "TrapClose [Quit.Win]"
 Wait
 [Quit.Win]
 Close #Win
 End
 [Link1.Click]
 Wait
 [Link1.Move]
 Wait

    [SetupWindow]
        NoMainWin
        WindowWidth = 400
        WindowHeight = 200
        UpperLeftX = Int((DisplayWidth-WindowWidth)/2)
        UpperLeftY = Int((DisplayHeight-WindowHeight)/2)

        Link1.Width = 80
        Link1.Height = 26

        GraphicBox #Win.Link1, 4, 4, Link1.Width, Link1.Height
        Stylebits #Win.Link1, 0, _WS_BORDER, 0, 0

        Open "Hyperlink Custom Control" For Window As #Win

        #Win, "TrapClose [Quit.Win]"

                Link1$ = cCc.Hyperlink$("#Win.Link1", Link1.Width
, Link1.Height, "Link 1 Text", _
        "Arial 10", "buttonface", "blue",
```

```
"[Link1.Click]", "[Link1.Move]")
    Wait

    [Quit.Win]
        Close #Win
    End

    [Link1.Click]
        Notice "You clicked Link 1."
    Wait

    [Link1.Move]
        Call cC.RollHyperlink Link1$, MouseX, MouseY
    Wait


    Function cCc.Hyperlink$(gbHndl$, width
, height, text$, font$, backcolor$, linkcolor$, eventClick$, eventMove
$)


#gbHndl$, "CLS; Down; Fill ";backcolor$ 'Clear the graphicbox and fill
 it with the specified backcolor.


'We need the height of the current font in pixels. Here's an easy way:
            #gbHndl$, "Place -100 -100 " 'Place pen offscreen
            #gbHndl$, "| " 'Print a blank line.

#gbHndl$, "PosXY penX penY" 'Get the coordinates of the pen. After dra
wing text, the pen moves down the proper

'height to make room for another line. Just subtract the end position
from the start position to get the height
                'of the font:
            fontHeight = penY-(-100)
'And it works, if you print the results out!

'-------------------------------------------------------------------'


#gbHndl$, "CLS; Down; Fill ";backcolor$;"; Flush" 'Redo the intial thi
ng just in case and this time flush it
            'so the graphics will stick.

#gbHndl$, "Font ";font$;" ; Color ";linkcolor$;" ; BackColor ";backcol
```

```
or$ 'Set the font, foreground and background
            'color.


'While we're at it, let's get the width of the string (so we can cente
r our text in the graphicbox!):
            #gbHndl$, "StringWidth? text$ stringWidth"
            'Calculate the x coordinate for text placement:
            Xcoord = Int((width-stringWidth)/2)

'-----------------------------------------------------------------'


#gbHndl$, "Place ";Xcoord;" ";fontHeight+2 'Set the pen at the proper
location so the text will draw correctly
            'in the graphicbox.

'Notice the text will start at the far left of the gbox, just like sta
tictext would.

        #gbHndl$, "|"; text$ 'Draw the text!


#gbHndl$, "Flush DefaultText" 'Make this drawing stick. And, give this
 drawing (segment) the name DefaultText .


#gbHndl$, "when leftButtonUp "; eventClick$ 'Set the graphicbox to jum
p to the branch label/sub that the user
            'specified when the link/graphicbox is clicked.
        #gbHndl$, "when mouseMove "; eventMove$


'Now, return the handle to a hyperlink so the user can pass it to the
hyperlink functions:

        hlinkActive = 0 'The hyperlink isn't active just yet.


'Return all the info as one big string of data seperated by spaces. We
'll parse this using word$() in another

'function. This allows the user to have more or less a "handle" to a l
ink.

        font$ = Sys.ReplaceChar$(font$," ","ø")
```

```
'Replace any spaces in the font text with a weird symbol, so we can pa
rse the
            'handle with word$() in another function.

        text$ = Sys.ReplaceChar$(text$," ","ø")
'Do the same with the text of the hyperlink. We'll reverse this in the
 other

'function using Sys.ReplaceChar$(text$,"_"," ") which will replace the
 underscores with spaces - so we'll be back
            'to normal. Same with the font$.

        cCc.Hyperlink$=gbHndl$;" ";width;" ";height;" "
;hlinkActive;" ";font$;" ";backcolor$;" ";linkcolor$;" ";_
        eventClick$;" ";eventMove$;" ";text$;" ";fontHeight

        'Word 1 = handle to the graphicbox.
        'Word 2 = the width of the graphicbox
        'Word 3 = the height of the graphicbox.

'Word 4 = 0/1 - whether or not the link is active (mouse over) hlinkAc
tive
        'Word 5 = the font used
        'Word 6 = the backcolor
        'Word 7 = the link color

'Word 8 = the event in which to trigger when the link is clicked.

'Word 9 = the event in which to trigger when the mouse is moved over t
he gbox.
        'Word 10 = the text for the hyperlink
        'Word 11 = the height of the font

    End Function

    Sub cC.RollHyperlink byref Link$, X, Y
        'Extract information from the hyperlink's handle:
        gbHndl$ = Word$(Link$,1) : width = Val(Word$(Link$,2))
 : height = Val(Word$(Link$,3))
        hlinkActive = Val(Word$(Link$,4))
 : font$ = Word$(Link$,5) : backcolor$ = Word$(Link$,6)
        linkcolor$ = Word$(Link$,7)
 : eventClick$ = Word$(Link$,8) : eventMove$ = Word$(Link$,9)
        text$ = Word$(Link$,10) : fontHeight = Val(Word$(Link$,11))
```

```
#gbHndl$, "Font ";font$;" ; Color ";linkcolor$;" ; BackColor ";backcol
or$ 'Make sure the colors and fonts are set
            'right...


'Now, replace the weird symbol in the font$ and text$ with spaces:
        font$ = Sys.ReplaceChar$(font$,"ø"," ")
        text$ = Sys.ReplaceChar$(text$,"ø"," ")

        'Recalculate the width of the text for placement's sake:
            #gbHndl$, "StringWidth? text$ stringWidth"
            'Calculate the x coordinate for text placement:
            Xcoord = Int((width-stringWidth)/2)

'-----------------------------------------------------------------'


'Check to see if the mouse coordinates (X, Y) are over the link's text
 - if so, we'll make sure the hyperlink
        'becomes underlined:


'First, let's get the width of the hyperlink's text using the stringwi
dth? command:
        #gbHndl$, "StringWidth? text$ textWidth"
        textWidth = textWidth + 1

'Now, we know both the width and height of the text. they are stored i
n textWidth and fontHeight . Perfect!


'Now, check to see if the mouse is over the actual area the text is dr
awn in:
            If X<=(textWidth+Xcoord)
 And X>=Xcoord And Y<=(fontHeight+2) And Y>=2 Then
'If the mouse is in the proper area:
                If Not(hlinkActive) Then

'If the link has not been drawn in it's mouseOver (active) state, then
 let's do so now.
                    'If hlinkActive was true (set to 1),

'it means we had already drawn it in it's active state, so we wouldn't
 need to redraw the same thing.
                    'That just causes flickering.
```

```
#gbHndl$, "DELSEGMENT DefaultText" 'Delete the inactive (normal) drawi
ng of the text.

#gbHndl$, "REDRAW" 'Update the graphicbox to reflect our deletion.

#gbHndl$, "Place ";Xcoord;" ";fontHeight+2 'Position the pen to redraw
 the text.

#gbHndl$, "Font ";font$;" underscore" 'Set the font to be underlined!
Of course,

'if the user already has an underlined font, this is a bummer. :(

#gbHndl$, "|";text$ 'Draw the new, underlined text!

#gbHndl$, "Flush ActiveText" 'Make this drawing stick, and call it Act
iveText -

'we can remove this drawing and redraw the old one when the mouse isn'
t
                        'over the graphicbox text area. Cool, huh?
                    hlinkActive = 1
'The hyperlink is now in it's active state!
                End If
            Else
'If the mouse is NOT over the actual text area, then let's make sure t
hat the hyperlink is drawn in it's
                 'inactive state.
                If hlinkActive Then
'If the hyperlink is being drawn in it's active state, then let's unac
tivate it!

#gbHndl$, "DELSEGMENT ActiveText" 'Remove the drawing of the active te
xt.
                    'Draw the text in it's normal state:
                    #gbHndl$, "REDRAW"
                    #gbHndl$, "Font ";font$
                    #gbHndl$, "Place ";Xcoord;" ";fontHeight+2
                    #gbHndl$, "|";text$

#gbHndl$, "Flush DefaultText" 'Make it stick and call this drawing the
 DefaultText drawing.
                    hlinkActive = 0 'hlinkActive is now inactive.
                End If
            End If
```

```
'-------------------------------------------------------------------
-------'

        'Replace spaces with underscores...
        font$ = Sys.ReplaceChar$(font$," ","ø")
        text$ = Sys.ReplaceChar$(text$," ","ø")


'Change the properties of the hyperlink's handle to match the updated
properties.

'This will actually affect the user's handle, because we passed Link$
by reference. (byref in the help file)
        Link$=gbHndl$;" ";width;" ";height;" "
;hlinkActive;" ";font$;" ";backcolor$;" ";linkcolor$;" ";_
        eventClick$;" ";eventMove$;" ";text$;" ";fontHeight

    End Sub

    'This is a helper function for the hyperlink functions.
    Function Sys.ReplaceChar$(String
$, FindChar$, ReplaceChar$) 'Find the character FindChar$ and replace
        'it with ReplaceChar$
        For i = 1 To Len(String$)
            char$=Mid$(String$,i,1)
            If char$=FindChar$ Then char$=ReplaceChar$
            Sys.ReplaceChar$=Sys.ReplaceChar$;char$
        Next i
    End Function

Link1$ = cCc.Hyperlink$("#Win.Link1", "Link 1 Text",
"Arial 10", "buttonface", "blue", "[Link1.Click]", "[Link1.Move]")

Link1$ = cCc.Hyperlink$("#Win.Link1", Link1.Width, Link1.Height
"Link 1 Text", "Arial 10", "buttonface", "blue",
"[Link1.Click]", "[Link1.Move]")
Function cCc.Hyperlink$(  gbHndl$,       width
,        height,      text$,       font$,     backcolor$, linkcolor$,
eventClick$,    eventMove$)

    [Link1.Move]
        Call cC.RollHyperlink Link1$, MouseX, MouseY
    Wait


        Link$=gbHndl$;" ";width;" ";height;" "
```

```
;hlinkActive;" ";font$;" ";backcolor$;" ";linkcolor$;" ";_
        eventClick$;" ";eventMove$;" ";text$;" ";fontHeight


#Win.Link1 80 26 1 Arialø10 buttonface blue [Link1.Click] [Link1.Move]
Linkø1øText 16
#Win.Link1 80 26 0 Arialø10 buttonface blue [Link1.Click] [Link1.Move]
Linkø1øText 16




Sub cC.RollHyperlink byref Link$, X, Y

Link1$ = cCc.Hyperlink$("#Win.Link1", Link1.Width, Link1.Height, "Link
 1 Text", "Arial 10", "buttonface", "blue", "[Link1.Click]", "[Link1.M
ove]")

    [Link1.Move]
        Call cC.RollHyperlink Link1$, MouseX, MouseY
    Wait


Global Link1$
```

You would also have to pass the name of the sub in for the mouse over event when you use the cCc.Hyperlink$() function, instead of the name of a branch label:

```
Link1$ = cCc.Hyperlink$("#Win.Link1", Link1.Width, Link1.Height, "Link
1 Text", "Arial 10", "buttonface", "blue", "[Link1.Click]",
"Link1.Move")




    Sub Link1.Move GraphicBox$, X, Y
            Call cC.RollHyperlink Link1$, X, Y
    End Sub

    NoMainWin
    WindowWidth = 600
    WindowHeight = 400
    Button #Win.b, "Unstep progress bar",[UnStepIt],UL,4,38,130,30
    Button #Win.b2, "Step progress bar",[StepIt],UL,138, 38, 130, 30
    Button #Win.b3, "Set Percentage",[Percent],UL,4,72,130,30
```

```
    GraphicBox #Win.g, 4, 4, 436, 30
    Stylebits #Win.g, 0,_WS_BORDER,0,0
    Open "Temp" For Window_NF As #Win
    #Win, "TrapClose [Quit]"
    #Win, "Font Arial 10"
    Prog1$ = cCc.ProgressBar$("#Win.g", 436, 30)
    Wait

    [UnStepIt]
        Call cC.UnStepProgressBar Prog1$
    Wait

    [StepIt]
        Call cC.StepProgressBar Prog1$
    Wait

    [Percent]
        Prompt "Choose percentage (0-100):"; Percent
        Call cC.SetProgressBarPercent Prog1$, Percent
    Wait

    [Quit]
    Close #Win
    End


'********************************************************************
*****************************************'
'BEGIN PROGRESS BAR FUNCTIONS ***************************************
*****************************************'
'********************************************************************
*****************************************'
    Function cCc.ProgressBar$(gbHndlb$, width, height)
'Create Custom Control Progress Bar
        gbHndl$ = Word$(gbHndlb$,1)

'Border drawing...sad this takes up a ton of code :( - but looks beaut
iful!
        dkGray$ = "102 102 102" : mdGray$ = "185 185 185"
 : ltGray$ = "234 234 234"

#gbHndl$, "CLS;Down;Fill White ; Size 1 ; Color ";ltGray$;"; Set 0 0 ;
 Color ";dkGray$;"; Line 1 0 ";(width-3);_
        " 0 ; Color 124 124 124 ; set ";(width-3);" 0"

#gbHndl$, "Color 167 167 170 ; Set ";(width-2);" 0 ; Color 178 178 178
```

```
 ; Set 0 1 ; Color 127 127 127 ; Set 0 2"

#gbHndl$, "Color ";mdGray$;" ; Line 1 1 ";(width-2);" 1 ; Color 117 11
7 117 ; Set ";(width-2);" 1"

#gbHndl$, "Color 167 167 170 ; Set ";(width-1);" 1 ; Color 124 124 124
 ; Set ";(width-1);" 2"

#gbHndl$, "Color ";ltGray$;" ; Line 1 2 ";(width-2);" 2 ; Color ";mdGr
ay$;";Set ";(width-3);" 1 ; Set ";(width-2);" 2"

#gbHndl$, "Set ";(width-3);" 2 ; Color ";dkGray$;" ; Line 0 3 0 ";(hei
ght-3)

#gbHndl$, "Color 124 124 124 ; Set 0 ";(height-3);" ; Color 167 167 16
7 ; Set 0 ";(height-2)

#gbHndl$, "Color ";dkGray$;" ; Line 1 ";(height-1);" ";(width-3);" ";(
height-1);";Line ";(width-1);" 3 ";(width-1);" ";(height-3)

#gbHndl$, "Color 124 124 124 ; Set ";(width-1);" ";(height-3);"; Set "
;(width-2); " ";(height-2);";Set ";(width-3);" ";(height-1)

#gbHndl$, "Color 167 167 167 ; Set ";(width-1);" ";(height-2);"; Set "
;(width-2); " ";(height-1);";Color ";ltGray$;_
        "; Line 1 ";(height-2);" ";(width-2);" ";(height-2)

#gbHndl$, "Color ";mdGray$;"; Line ";(width-2);" 2 ";(width-2);" ";(he
ight-2);";Color ";ltGray$;"; Line ";(width-3);" 3 ";_
        (width-3);" ";(height-2)
        barWidth = Int(width/50)
        padWidth = Int(width/218)
        maxBars = Int(width/(barWidth+padWidth))
'holds how many bars this progressbar holds...
        cCc.ProgressBar$ = gbHndl$;" ";width;" ";height;" "
;barWidth;" ";padWidth;" ";barCount;" ";maxBars
        #gbHndl$, "backcolor 255 255 255"
        #gbHndl$, "Flush"
    End Function
    Sub cC.SetProgressBarPercent byref progHndl$, percent
        gbHndl$=Word$(progHndl$,1):width=Val(Word$(
progHndl$,2)):height=Val(Word$(progHndl$,3)):barWidth=Val(Word$(
progHndl$,4))
        padWidth=Val(Word$(progHndl$,5)):barCount=Val(Word$(
progHndl$,6)):maxBars=Val(Word$(progHndl$,7))
        p = Int((maxBars/100)*percent)
```

```
        If p = 0 Then
            progHndl$=cCc.ProgressBar$(gbHndl$;" .", width, height)
            Exit Sub
        End If
        If p>barCount Then
            dif = p-barCount
            For i = 1 To dif
                Call cC.StepProgressBar progHndl$
            Next i
        End If
        If p<barCount Then
            dif = barCount - p
            For i = 1 to dif
                Call cC.UnStepProgressBar progHndl$
            Next i
        End If
        progHndl$ = progHndl$
    End Sub
    Sub cC.StepProgressBar byref progHndl$
        gbHndl$=Word$(progHndl$,1):width=Val(Word$(
progHndl$,2)):height=Val(Word$(progHndl$,3)):barWidth=Val(Word$(
progHndl$,4))
        padWidth=Val(Word$(progHndl$,5)):barCount=Val(Word$(
progHndl$,6)):maxBars=Val(Word$(progHndl$,7))
        barCount=barCount+1
        If barCount>maxBars Then Exit Sub
        If barCount>1 Then Lx=((barWidth+padWidth)*(barCount-1))+1
Else Lx=1
        LxF = Lx+barWidth

#gbHndl$, "Color 181 200 226;Line ";Lx;" 1 ";LxF;" 1;Color 164 191 221
;Line ";Lx;" 2 ";LxF;" 2;Color 146 180 219;Line ";_
        Lx;" 3 ";LxF;" 3"

#gbHndl$, "Color 130 166 214;Line ";Lx;" 4 ";LxF;" 4;Color 114 156 211
;Line ";Lx;" 5 ";LxF;" 5;Color 95 145 206;Line ";_
        Lx;" 6 ";LxF;" 6"

#gbHndl$, "BackColor 87 138 204; Color 87 138 204; Size 1; Place ";Lx;
" 7; BoxFilled ";LxF;" ";(height-6)

#gbHndl$, "Color 90 142 206;Line ";Lx;" ";(height-6);" ";LxF;" ";(heig
ht-6);"; Color 104 151 209;Line ";Lx;" ";(height-5);" ";_
        LxF;" ";(height-5)

#gbHndl$, "Color 124 163 214;Line ";Lx;" ";(height-4);" ";LxF;" ";(hei
```

```
ght-4);";Color 140 173 216;Line ";Lx;" ";(height-3);" ";_
        LxF;" ";(height-3)

#gbHndl$, "Color 159 187 221;Line ";Lx;" ";(height-2);" ";LxF;" ";(hei
ght-2)
        progHndl$ = gbHndl$;" ";width;" ";height;" "
;barWidth;" ";padWidth;" ";barCount;" ";maxBars
        #gbHndl$, "Backcolor 255 255 255"
        #gbHndl$, "Flush bar";barCount
    End Sub
    Sub cC.UnStepProgressBar byref progHndl$
        gbHndl$=Word$(progHndl$,1):width=Val(Word$(
progHndl$,2)):height=Val(Word$(progHndl$,3)):barWidth=Val(Word$(
progHndl$,4))
        padWidth=Val(Word$(progHndl$,5)):barCount=Val(Word$(
progHndl$,6)):maxBars=Val(Word$(progHndl$,7))
        If barCount=0 Then Exit Sub
        barSeg = barCount' + 1
        #gbHndl$, "delsegment bar";barSeg
        #gbHndl$, "redraw"
        if barCount>0 Then barCount=barCount-1
        progHndl$ = gbHndl$;" ";width;" ";height;" "
;barWidth;" ";padWidth;" ";barCount;" ";maxBars
    End Sub
'*****************************************************************
*****************************************'
'END PROGRESS BAR FUNCTIONS ***********************************
*****************************************'
'*****************************************************************
*****************************************'
```