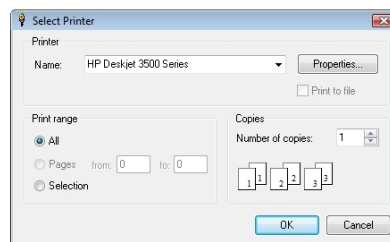


The following demo was submitted by Robert McAllister (- [robmcsl](#)). Use this code to incorporate a look alike printer dialog window in your program. It is setup to work with any DPI setting but that portion can be easily removed if desired. Also, since these functions only work on 2000/XP on up, it checks what platform the program is running on.

It has also been noted by Stefan Pendl (- [stpendl](#)) that there is no need to scale native LB controls in a dialog window, since dialog windows are automatically scaled to dpi by Liberty BASIC. API created controls must be scaled manually, even in a dialog window.



```
DefaultPrinterName$=DefaultPrinterName$()
gosub [SelectPrinter]
end
```

```
[SelectPrinter]
  WinVersion=WinVersion()
  if WinVersion>=5 then gosub [
Find2000xpvistaprinters]'find installed printers
  if WinVersion<=4 then notice "Incompatible OS"+chr$(13)+_
"This function will only work on Windows 2000,"+chr$(13)+_
  "Windows XP or Windows Vista":wait
  TEMPSelectedPrinter$=SelectedPrinter$

  ScrnScl=GetScaleFactor()

  WindowWidth=int(440/ScrnScl):WindowHeight=int(270/ScrnScl)
  UpperLeftX=int(((DisplayWidth-WindowWidth)/2)/ScrnScl)
  UpperLeftY=int(((DisplayHeight-WindowHeight)/2)/ScrnScl)
  BackgroundColor$ = "buttonface"

  groupbox #pselect.one, "Printer",      int(13/ScrnScl),  int(6/
ScrnScl),  int(410/ScrnScl),  int(80/ScrnScl)
```

```
groupbox #pselect.two, "Print range", int(13/Scrnscl), int(92/
Scrnscl), int(217/Scrnscl), int(105/Scrnscl)
groupbox #pselect.thr, "Copies", int(240/Scrnscl), int(92/
Scrnscl), int(183/Scrnscl), int(105/Scrnscl)

statictext #pselect.1, "Name:", int(25/Scrnscl), int(33/
Scrnscl), int(55/Scrnscl), int(26/Scrnscl)
statictext #pselect.6, "from:", int(89/Scrnscl), int(145/
Scrnscl), int(26/Scrnscl), int(23/Scrnscl)
statictext #pselect.7, "to:", int(165/Scrnscl), int(
145/Scrnscl), int(12/Scrnscl), int(23/Scrnscl)

statictext #pselect.8, "Number of copies:", int(251/
Scrnscl), int(115/Scrnscl), int(112/Scrnscl), int(23/Scrnscl)

combobox #pselect.list1, prnt$(),[pSelected1], int(80/
Scrnscl), int(29/Scrnscl), int(227/Scrnscl), 100

button #pselect.default,"OK",[pSelected2], UL,
int(270/Scrnscl), int(207/Scrnscl), int(75/Scrnscl),
int(24/Scrnscl)
button #pselect.Cancel,"Cancel",[quit.pselect], UL, int(
351/Scrnscl), int(207/Scrnscl), int(75/Scrnscl), int(24/Scrnscl)
button #pselect.Cancel,"Properties...",[pProperties], UL,int(
319/Scrnscl), int(26/Scrnscl), int(92/Scrnscl), int(24/Scrnscl)

checkbox #pselect.checkbox1, "Print to file", [pyes], [pno],int(
319/Scrnscl), int(57/Scrnscl), int(80/Scrnscl), int(25/Scrnscl)

radiobutton #pselect.one, "All", [pyes], [pno], int(25/
Scrnscl), int(115/Scrnscl), int(60/Scrnscl), int(20/Scrnscl)
radiobutton #pselect.two, "Pages", [pyes], [pno], int(25/
Scrnscl), int(140/Scrnscl), int(60/Scrnscl), int(20/Scrnscl)
radiobutton #pselect.thr, "Selection", [pyes], [pno], int(25/
Scrnscl), int(165/Scrnscl), int(85/Scrnscl), int(20/Scrnscl)
textbox #pselect.textbox1, int(116/Scrnscl), int(140/
Scrnscl),int(38/Scrnscl),int(20/Scrnscl)
textbox #pselect.textbox2, int(180/Scrnscl), int(140/
Scrnscl),int(38/Scrnscl),int(20/Scrnscl)

textbox #pselect.textbox3, int(363/Scrnscl), int(110/
Scrnscl),int(48/Scrnscl),int(20/Scrnscl)

stylebits #pselect.g1, 0, _WS_BORDER, 0, 0
stylebits #pselect.g2, 0, _WS_BORDER, 0, 0
stylebits #pselect.g3, 0, _WS_BORDER, 0, 0
```

```

    stylebits #pselect.g4, 0, _WS_BORDER, 0, 0
    stylebits #pselect.g5, 0, _WS_BORDER, 0, 0
    stylebits #pselect.g6, 0, _WS_BORDER, 0, 0
    graphicbox #pselect.g4, int(252/ScrnScl),int(155/ScrnScl) , int(
20/ScrnScl), int(26/ScrnScl)
    graphicbox #pselect.g5, int(289/ScrnScl),int(155/ScrnScl) , int(
20/ScrnScl), int(26/ScrnScl)
    graphicbox #pselect.g6, int(326/ScrnScl),int(155/ScrnScl) , int(
20/ScrnScl), int(26/ScrnScl)
    graphicbox #pselect.g1, int(264/ScrnScl),int(148/ScrnScl) , int(
20/ScrnScl), int(26/ScrnScl)
    graphicbox #pselect.g2, int(302/ScrnScl),int(148/ScrnScl) , int(
20/ScrnScl), int(26/ScrnScl)
    graphicbox #pselect.g3, int(338/ScrnScl),int(148/ScrnScl) , int(
20/ScrnScl), int(26/ScrnScl)

```

```

open "Select Printer" for dialog_modal as #pselect
#pselect "trapclose [quit.pselect]"
#pselect.list1 "select ";TEMPSelectedPrinter$

```

```

'create spinner control:http://libertybasic.conforums.com/index.cgi?bo
ard=LB3&action=display&num=1217524188

```

```

    UpperLimit = 200
    LowerLimit = 1
    CurrentPosition = 1
    hWnd = hWnd(#pselect)
    hText = hWnd(#pselect.textbox3)
    UDS.ARROWKEYS = hexdec("20")
    UDS.WRAP = hexdec("1")
    UDS.SETBUDDYINT = hexdec("2")
    UDS.ALIGNRIGHT = hexdec("4")
    dwStyle = _WS_CHILD or _WS_BORDER or _WS_VISIBLE or
UDS.ARROWKEYS or UDS.WRAP or UDS.SETBUDDYINT or UDS.ALIGNRIGHT
    hSpinner = CreateSpinner(hWnd, dwStyle, hText,
UpperLimit, LowerLimit, CurrentPosition)

```

```

#pselect.checkbox1 "disable"
#pselect.two "disable"
#pselect.6 "!disable"
#pselect.7 "!disable"
#pselect.textbox1 "0"
#pselect.textbox2 "0"
#pselect.textbox1 "!disable"
#pselect.textbox2 "!disable"
#pselect.one "set"

```

```
#pselect.default "!setfocus"

for x=1 to NumberOfPrinters
  if prnt$(x)=DefaultPrinterName$ then #pselect.list1,
"selectindex ";x
next x

for x=1 to 6
  c=c+1:var$ = "#pselect.g";x
  #var$, "font Times_New_Roman 7"
  #var$, "down;color darkgray"
  #var$, "line 0 0 30 0"
  #var$, "line 0 0 0 25"
  #var$, "color black"
  #var$, "line 19 0 19 25"
  #var$, "line 0 24 20 24"
  #var$, "place 10 21"
  #var$, "\";c
  #var$, "flush"
  if c=3 then c=0
next x

wait

[pProperties]
  notice "This feature is not available."
  wait

[pyes]
  wait

[pno]
  wait

[quit.pselect]
  SelectedPrinter$=TEMPSelectedPrinter$
  callDll #user32, "DestroyWindow",_
    hSpinner as ulong, ret as long
  close #pselect
  end

[pSelected1]
  #pselect.list1 "selection? sel$"
  SelectedPrinter$=sel$
  wait
```

```
[pSelected2]
#pselect.list1 "selection? sel$"
SelectedPrinter$=sel$
if sel$="" then notice "Please select a printer"
calldll #user32, "DestroyWindow",_
    hSpinner as ulong, ret as long
print "Selected printer...";SelectedPrinter$
#pselect.textbox3 "!contents? copies"
print "Number of copies...";copies
close #pselect
end

[setdefaultprinter]
open "winspool.drv" for dll as #winspool
calldll #winspool, "SetDefaultPrinterA",_
    Printer$ as ptr,_
    SetDefaultPrinter as long
close #winspool
if SetDefaultPrinter=0 then call DisplayError
return

[Find2000xpvistaprinters]
'find printers
PRINTER.ENUM.LOCAL = hexdec("2")
PRINTER.ENUM.CONNECTIONS = hexdec("4")
nLocalPrinters = EnumPrinters(PrinterInfo$, PRINTER.ENUM.LOCAL)
LocalPrintersInfo$ = PrinterInfo$
PrinterInfo$ = ""
nNetworkPrinters = EnumPrinters(PrinterInfo$,
PRINTER.ENUM.CONNECTIONS)
NetworkPrintersInfo$ = PrinterInfo$
PrinterInfo$ = ""
NumberOfPrinters = nLocalPrinters + nNetworkPrinters
PrinterInfo$ = LocalPrintersInfo$;"";NetworkPrintersInfo$
dim prnt$(NumberOfPrinters)
if NumberOfPrinters then 'remove faxes from list
    prnt=0
    for x1=1 to NumberOfPrinters
'remove "faxes", "Quicken pdf creator" and "Intuit pdf creator" from l
ist
        if instr(lower$(word$(PrinterInfo$,x1,";")), "fax")=0 _
            and instr(lower$(word$(PrinterInfo$,x1,";")), "quick")=0 _
            and instr(lower$(word$(PrinterInfo$,x1,";")), "intuit")=0 _
            then prnt=prnt+1:prnt$(prnt)=word$(PrinterInfo$,x1,";")
    next x1
end if
```

```
return
```

```
function EnumPrinters(byref PrinterInfo$, pLocale)
    'returns the number of printers found
    'fills the submitted variable with the printer names
    'separated by semicolons (;)
    open "winspool.drv" for dll as #winspool
    struct pcbNeeded, value as ulong
    struct pcReturned, value as ulong
    struct PrinterInfo4,_
        pPrinterName$ as ptr,_
        pServerName$ as ptr,_
        Attributes as ulong
    PrinterInfo4Len = len(PrinterInfo4.struct)
    nFlags = pLocale
    Level = 4
    cbBuf = PrinterInfo4Len
    uFlags = _LMEM_MOVEABLE or _LMEM_ZEROINIT
    calldll #kernel32, "LocalAlloc", _
        uFlags as ulong, _
        cbBuf as ulong, _
        hMem as ulong
    calldll #kernel32, "LocalLock", _
        hMem as ulong, _
        pBuffer as ulong
```

```
[retryEnumPrinters]
    calldll #winspool, "EnumPrintersA",_
        nFlags as ulong,_
        PrinterName as ulong,_
        Level as ulong,_
        pBuffer as ulong,_
        cbBuf as ulong,_
        pcbNeeded as struct,_
        pcReturned as struct,_
        result as long
    if result = 0 then
        if GetLastError() = 122 then
            cbBuf = pcbNeeded.value.struct
            hOldMem = hMem
            calldll #kernel32, "LocalReAlloc", _
                hOldMem as ulong, _
                cbBuf as ulong, _
                uFlags as ulong, _
                hMem as ulong
            calldll #kernel32, "LocalLock", _
```

```
        hMem as ulong, _
        pBuffer as ulong
    goto [retryEnumPrinters]
else
    call DisplayError
end if
else
    EnumPrinters = pcReturned.value.struct
    EnumPrinters$(1)=str$(EnumPrinters)
    BufferPointer = pBuffer
    for count = 0 to EnumPrinters - 1
        calldll #kernel32, "RtlMoveMemory", _
            PrinterInfo4 as struct, _
            BufferPointer as ulong, _
            PrinterInfo4Len as ulong, _
            result as void
        BufferPointer = BufferPointer + PrinterInfo4Len
        pointer = PrinterInfo4.pPrinterName$.struct
        PrinterInfo$ = winstring(pointer); ";"; PrinterInfo$
    next
    PrinterInfo$ = left$(PrinterInfo$, len(PrinterInfo$)-1)
end if
calldll #kernel32, "LocalFree", _
hMem as ulong, _
result as ulong
close #winspool
end function

function GetLastError()
    calldll #kernel32, "GetLastError", _
    GetLastError as ulong
end function

sub DisplayError
    dwFlags = _FORMAT_MESSAGE_FROM_SYSTEM
    nSize = 1024
    lpBuffer$ = space$(nSize); chr$(0)
    dwMessageID = GetLastError()
    calldll #kernel32, "FormatMessageA", _
        dwFlags as ulong, _
        lpSource as ulong, _
        dwMessageID as ulong, _
        dwLanguageID as ulong, _
        lpBuffer$ as ptr, _
        nSize as ulong, _
        Arguments as ulong, _
```

```
        result as ulong
    if trim$(lpBuffer$)<>
"The operation completed succes
sfully." then notice "Error "; dwMessageID; chr$(13); left$(
lpBuffer$, result)
end sub
```

```
function DefaultPrinterName$()
    struct pcchBuffer, value as ulong
    pszBuffer$ = space$(500)
    pcchBuffer.value.struct = 500
    open "winspool.drv" for dll as #winspool
    calldll #winspool, "GetDefaultPrinterA", _
        pszBuffer$ as ptr, _
        pcchBuffer as struct, _
        GetDefaultPrinter as long
    close #winspool
    DefaultPrinterName$ = left$(pszBuffer$,
pcchBuffer.value.struct - 1)
end function
```

```
function WinVersion()
    struct OSVERSIONINFO, _
        dwOSVersionInfoSize as ulong, _
        dwMajorVersion as ulong, _
        dwMinorVersion as ulong, _
        dwBuildNumber as ulong, _
        dwPlatformId as ulong, _
        szCSDVersion as char[128]

    L=len(OSVERSIONINFO.struct)
    OSVERSIONINFO.dwOSVersionInfoSize.struct=L
    calldll #kernel32, "GetVersionExA", _
        OSVERSIONINFO as struct, _
        result as long
    WinVersion=OSVERSIONINFO.dwMajorVersion.struct
'dwMajorVersion
'Identifies the major version number of the operating
'system as follows.
'Windows 95                4
'Windows 98                4
'Windows Me                4
'Windows NT 3.51           3
'Windows NT 4.0            4
'Windows 2000              5
'Windows XP                5
```

```
'Windows .NET Server      5
'Windows Vista            6
'Windows 7                6
end function

function GetScaleFactor()
    LOGPIXELSX = 88
    nIndex = LOGPIXELSX

    calldll #user32, "GetDC", _
        0 as ulong, _ ' entire screen
        hDC as ulong

    calldll #gdi32, "GetDeviceCaps", _
        hDC as ulong, _
        nIndex as ulong, _
        dpi as ulong

    calldll #user32, "ReleaseDC", _
        0 as ulong, _ ' entire screen
        hDC as ulong, _
        result as long ' 1 = success

    GetScaleFactor = dpi / 96
end function

Function CreateSpinner(hWnd, style, hTxt, upr, lwr, pos)
    CallDLL #comctl32, "InitCommonControlsEx", _
        re as void

    CallDLL #user32, "GetWindowLongA", _
        hWnd as ulong, _
        _GWL_HINSTANCE as long, _
        hInst as ulong

    CallDLL #comctl32, "CreateUpDownControl", _
        style as ulong, _
        x as long, _
        y as long, _
        w as long, _
        h as long, _
        hWnd as ulong, _
        id as long, _
        hInst as ulong, _
        hTxt as ulong, _
        upr as long, _
```

```
    lwr as long,_  
    pos as long,_  
    CreateSpinner as ulong  
End Function
```