

EDIT:

This is the final version for the Lite edition of LB Help Search

Version 1.03b for discussion only

changed createBrowseControl from subroutine to called sub

Reason: working towards using handle\$ to handle button clicks

Would like comment on whether the calls and the sub are properly formed...

```
' LB Help Search entry
' Author: jaba
' Date: May 5, 2011
'
' History:
' v1.0      May 6, 2011      Submitted
' v1.01     May 6, 2011      Added browser control
' v1.02     May 9, 2011      Added GetModuleFileName$() routine
' v1.03     May 12, 2011     Additional pages added to quick viewer
'                                     Some names made more purpose-
descriptive
'

' v1.03b    May 13, 2011     in progress...using sub to create ATL brow
ser window
'
'
' This code is free for personal use.
'
' Based on a framework provided by Alyce on LB forum.
'
' Some parts taken from entries by others, including
' Alyce, Janet, Stefan, and Gordon.
'
' Thanks to them and others.
'

' This is a basic and minimum interface which uses the default browser
' to display help file pages.
'
'
' Thanks to Stefan Pendl for API function method to obtain location of
' LB installation directory. This method is more reliable with systems
```

```
' running Vista/Win7 than is the StartupDir$ or DefaultDir$ method.
InstallDir$ = GetFolder$(GetModuleFileName$())
HelpDir$ = InstallDir$ + "lb4help\LibertyBASIC_4_web\HTML\"
HelpFileIndex$ = InstallDir$; "lb4help\LibertyBASIC_4.html"

'List of Quick View pages:
'Quick Start Guide  = "libe3mnn.htm"
'Command Reference  = "libe2co3.htm"
'Reserved Word List = "libe6ldg.htm"

Open "atl" For DLL As #atl

CallDLL #atl, "AtlAxWinInit", Ret As long

nomainwin
WindowWidth  = 442
WindowHeight = 520
UpperLeftX   = DisplayWidth - WindowWidth
UpperLeftY   = 40

statictext #s.search "Search for: ", 10, 15, 200, 20
textbox #s.term 10, 35, 320, 25
button #s.content "Contents", [openContentsPage], UL, 340, 5, 80,
25
button #s.default "Find", [findPages], UL, 340, 35, 80, 25
listbox #s.pages, found$(), [
displaySelectedPage], 10, 65, 410, 190

button #s.quick "Quick Start Guide", [quickButton], UL, 100,
260, 100, 25 'points to libe3mnn.htm
button #s.commands "Command Reference", [
commandsButton], UL, 205, 260, 115, 25 'points to libe2co3.htm
button #s.reserve "Reserved Words", [reservedButton], UL, 325,
260, 95, 25 'points to libe6ldg.htm

statictext #s.heading "Quick Viewer", 10, 267, 90, 20
graphicbox #s.resource 10, 290, 410, 170
'container for browser control
statictext #s.status, "Right-
click browser window for BACK/FORWARD.", 10, 468, 410, 20

open "Search Me!" for dialog_nf as #s
```

```
#s "trapclose [quit]"
```

```
#s "font ms_sans_serif 9"  
#s.content "!font ms_sans_serif 9 bold"  
#s.search  "!font ms_sans_serif 9 bold"  
#s.heading "!font ms_sans_serif 9 bold"  
#s.status  "!font arial 7"
```

```
'display Quick Start Guide in browse control  
QuickViewPage$ = "libe3mnn.htm"  
call createBrowseControl HelpDir$, QuickViewPage$, hFlag  
#s.term "!setfocus"
```

```
dim info$(10,10)  
files HelpDir$, "*.htm", info$()
```

```
numFiles = val(info$(0,0))
```

```
wait
```

```
[quickButton]
```

```
QuickViewPage$ = "libe3mnn.htm"  
call createBrowseControl HelpDir$, QuickViewPage$, hFlag  
wait
```

```
[commandsButton]
```

```
QuickViewPage$ = "libe2co3.htm"  
call createBrowseControl HelpDir$, QuickViewPage$, hFlag  
wait
```

```
[reservedButton]
```

```
QuickViewPage$ = "libe6ldg.htm"  
call createBrowseControl HelpDir$, QuickViewPage$, hFlag  
wait
```

```
[findPages]
```

```
#s.term "!contents? term$"
```

```
dim found$(numFiles)
```

```
for i = 1 to numFiles
```

```
    open HelpDir$ + info$(i,0) for input as #file  
    txt$ = INPUT$(#file, lof(#file))
```

```
        if instr(txt$, term$) >0 then
            found$(i) = GetTitle$(txt$); chr$(0); info$(i,0)
            #s.pages "reload"
        end if
    close #file

next
wait

[openContentsPage]
    'open user's default browser
    run
"rundll32.exe url.dll,FileP
rotocolHandler "; chr$(34); HelpFileIndex$; chr$(34)

    #s.term "!setfocus"
    wait

[displaySelectedPage]
    #s.pages "selection? FileName$"
    FileName$ = word$(FileName$, 2, chr$(0))

    'open in user's default browser or new tab
    run
"rundll32.exe url.dll,FileP
rotocolHandler "; chr$(34); HelpDir$; FileName$; chr$(34)

    wait

[quit]
    close #s
    close #atl
end

'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

'SUBS AND FUNCTIONS

'credit: Alyce's entry.
'extract title from html text
function GetTitle$(s$)
    marker1=instr(upper$(s$), "<A NAME=")
    marker1=instr(upper$(s$), "<B>",marker1)+3
    marker2=instr(upper$(s$), "</B>", marker1)
    lenTitle=marker2-marker1
```

```
GetTitle$=mid$(s$,marker1,lenTitle)
end function
```

```
'borrowed from Stefan Pendl
```

```
'added 5/9/11
```

```
function GetModuleFileName$()
    nSize = _MAX_PATH + 1
    lpFilename$ = space$(nSize); CHR$(0)
    callDll #kernel32, "GetModuleFileNameA", _
        hModule      as uLong, _
        lpFilename$ as ptr, _
        nSize        as uLong, _
        result       as uLong
    if result > 0 then GetModuleFileName$ = trim$(lpFilename$)
end function
```

```
function GetFolder$(Path$)
    pos = 1
    GetFolder$ = Path$
    while pos > 0
        pos = instr(Path$, "\", pos)
        if pos > 0 then
            GetFolder$ = left$(Path$, pos)
            pos = pos + 1
        end if
    wend
end function
```

```
sub createBrowseControl D$, F$, byref hATL
    if hATL then
        CallDLL #user32, "DestroyWindow", _
            hATL As ulong, _
            result As long
        hATL = 0
    end if

    hWndContainer = hWnd(#s.resource) 'graphicbox handle

    callDLL #user32, "GetWindowLongA", _
        hWndContainer as ulong, _
        _GWL_HINSTANCE as long, _
        hInst as ulong

    ResName$ = D$ + F$
```

```
style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL
CallDLL #user32, "CreateWindowExA", _
    _WS_EX_STATICEDGE As long, _      'extended type
    "AtIAxWin" As ptr, _               'class name
    ResName$ As ptr, _                 'resource file name & path
    style As long, _                   'window style
    0 As long, _                       'left x pos
    0 As long, _                       'top y pos
    408 As long, _                     'width
    168 As long, _                     'height
    hWndContainer As ulong, _          'handle of container
    100 As long, _
'handle to menu or child window ID
    hInst As ulong, _                  'parent instance handle
    0 As long, _                       'window creation data
    hATL As ulong
'handle of active template library control

end sub
```

Version 1.03 Additional pages added to quick viewer

```
' LB Help Search entry
' Author: jaba
' Date: May 5, 2011
'
' History:
' v1.0      May 6, 2011      Submitted
' v1.01     May 6, 2011      Added browser control
' v1.02     May 9, 2011      Added GetModuleFileName$() routine
' v1.03     May 12, 2011     Additional pages added to quick viewer
'                                     Some names made more purpose-
descriptive
'
'
' This code is free for personal use.
'
' Based on a framework provided by Alyce on LB forum.
'
' Some parts taken from entries by others, including
' Alyce, Janet, Stefan, and Gordon.
'
' Thanks to them and others.
'
```

```
' This is a basic and minimum interface which uses the default browser
' to display help file pages.
'
'

' Thanks to Stefan Pendl for API function method to obtain location of
' LB installation directory. This method is more reliable with systems
' running Vista/Win7 than is the StartupDir$ or DefaultDir$ method.
InstallDir$ = GetFolder$(GetModuleFileName$())
HelpDir$ = InstallDir$ + "lb4help\LibertyBASIC_4_web\HTML\"
HelpFileIndex$ = InstallDir$; "lb4help\LibertyBASIC_4.html"

'List of Quick View pages:
'Quick Start Guide = "libe3mnn.htm"
'Command Reference = "libe2co3.htm"
'Reserved Word List = "libe6ldg.htm"

Open "atl" For DLL As #atl

CallDLL #atl, "AtlAxWinInit", Ret As long

nomainwin
WindowWidth = 442
WindowHeight = 520
UpperLeftX = DisplayWidth - WindowWidth
UpperLeftY = 40

statictext #s.search "Search for: ", 10, 15, 200, 20
textbox #s.term 10, 35, 320, 25
button #s.content "Contents", [openContentsPage], UL, 340, 5, 80,
25
button #s.default "Find", [findPages], UL, 340, 35, 80, 25
listbox #s.pages, found$(), [
displaySelectedPage], 10, 65, 410, 190

button #s.quick "Quick Start Guide", [quickButton], UL, 100,
260, 100, 25 'points to libe3mnn.htm
button #s.commands "Command Reference", [
commandsButton], UL, 205, 260, 115, 25 'points to libe2co3.htm
button #s.reserve "Reserved Words", [reservedButton], UL, 325,
```

```
260, 95, 25          'points to libe6ldg.htm

    statictext #s.heading "Quick Viewer", 10, 267, 90, 20
    graphicbox #s.resource 10, 290, 410, 170
'container for browser control
    statictext #s.status, "Right-
click browser window for BACK/FORWARD.", 10, 468, 410, 20

    open "Search Me!" for dialog_nf as #s
    #s "trapclose [quit]"

    #s "font ms_sans_serif 9"
    #s.content "!font ms_sans_serif 9 bold"
    #s.search  "!font ms_sans_serif 9 bold"
    #s.heading "!font ms_sans_serif 9 bold"
    #s.status  "!font arial 7"

    'display Quick Start Guide in browse control
    QuickViewPage$ = "libe3mnn.htm"
    gosub [createBrowseControl]

    #s.term "!setfocus"

    dim info$(10,10)
    files HelpDir$, "*.htm", info$()

    numFiles = val(info$(0,0))

    wait

[quickButton]
    QuickViewPage$ = "libe3mnn.htm"
    gosub [createBrowseControl]
    wait

[commandsButton]
    QuickViewPage$ = "libe2co3.htm"
    gosub [createBrowseControl]
    wait

[reservedButton]
    QuickViewPage$ = "libe6ldg.htm"
    gosub [createBrowseControl]
    wait
```

```
[findPages]
    #s.term "!contents? term$"

    dim found$(numFiles)

    for i = 1 to numFiles

        open HelpDir$ + info$(i,0) for input as #file
        txt$ = INPUT$(#file, lof(#file))

        if instr(txt$, term$) >0 then
            found$(i) = GetTitle$(txt$); chr$(0); info$(i,0)
            #s.pages "reload"
        end if
        close #file

    next
    wait

[openContentsPage]
    'open user's default browser
    run
    "rundll32.exe url.dll,FileP
    rotocolHandler "; chr$(34); HelpFileIndex$; chr$(34)

    #s.term "!setfocus"
    wait

[displaySelectedPage]
    #s.pages "selection? FileName$"
    FileName$ = word$(FileName$, 2, chr$(0))

    'open in user's default browser or new tab
    run
    "rundll32.exe url.dll,FileP
    rotocolHandler "; chr$(34); HelpDir$; FileName$; chr$(34)

    wait

[quit]
    close #s
    close #atl
    end
```

```
'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
'SUBS AND FUNCTIONS
```

```
'credit: Alyce's entry.
```

```
'extract title from html text
```

```
function GetTitle$(s$)
    marker1=instr(upper$(s$), "<A NAME=")
    marker1=instr(upper$(s$), "<B>",marker1)+3
    marker2=instr(upper$(s$), "</B>", marker1)
    lenTitle=marker2-marker1
    GetTitle$=mid$(s$,marker1,lenTitle)
end function
```

```
'borrowed from Stefan Pendl
```

```
'added 5/9/11
```

```
function GetModuleFileName$()
    nSize = _MAX_PATH + 1
    lpFilename$ = space$(nSize); CHR$(0)
    callDll #kernel32, "GetModuleFileNameA", _
        hModule      as uLong, _
        lpFilename$ as ptr, _
        nSize        as uLong, _
        result        as uLong
    if result > 0 then GetModuleFileName$ = trim$(lpFilename$)
end function
```

```
function GetFolder$(Path$)
    pos = 1
    GetFolder$ = Path$
    while pos > 0
        pos = instr(Path$, "\", pos)
        if pos > 0 then
            GetFolder$ = left$(Path$, pos)
            pos = pos + 1
        end if
    wend
end function
```

```
[createBrowseControl]
    if hATL then
        CallDLL #user32, "DestroyWindow", _
            hATL As ulong, _
            result As long
```

```
        hATL = 0
    end if

    hWndContainer = hWnd(#s.resource)    'graphicbox handle

    callDLL #user32, "GetWindowLongA", _
        hWndContainer as ulong, _
        _GWL_HINSTANCE as long, _
        hInst as ulong

    ResName$ = HelpDir$ + QuickViewPage$

    style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL
    CallDLL #user32, "CreateWindowExA", _
        _WS_EX_STATICEDGE As long, _    'extended type
        "AtIAxWin" As ptr, _            'class name
        ResName$ As ptr, _              'resource file name & path
        style As long, _                'window style
        0 As long, _                    'left x pos
        0 As long, _                    'top y pos
        408 As long, _                  'width
        168 As long, _                  'height
        hWndContainer As ulong, _        'handle of container
        100 As long, _
        'handle to menu or child window ID
        hInst As ulong, _                'parent instance handle
        0 As long, _                    'window creation data
        hATL As ulong
        'handle of active template library control

    return
```

Version 1.02 added GetModuleFileName\$() routine

```
' LB Help Search entry
' Author: jaba
' Date: May 5, 2011
'
' History:
' v1.0      May 6, 2011      Submitted
' v1.01     May 6, 2011      Added browser control
' v1.02     May 9, 2011      Added GetModuleFileName$() routine
'
```

```
' This code is free for personal use.
'
' Based on a framework provided by Alyce on LB forum.
'
' Some parts taken from entries by others, including
' Alyce, Janet, Stefan, and Gordon.
'
' Thanks to them and others.
'

' This is a basic and minimum interface which uses the default browser
' to display search terms. As I learn more, I may include a browser co
ntrol

' in the interface, which I think is the best and fastest way to displ
ay

' the search terms. But, for now, I'm happy to be making a submission.
'

' To display the Reserved Word List, I edited the .htm file and saved
it

' as a .txt file in my /HTML/ sub folder. The user will see the html
' file in the lower textbox, which includes html tags. I have not made
a

' parser yet, or included the browser control to display html files.
' At this point, it's an idea only.
'

' Thanks to Stefan Pendl for API function method to obtain location of
' LB installation directory. This method is more reliable with systems
' running Vista/Win7 than is the StartupDir$ or DefaultDir$ method.
InstallDir$ = GetFolder$(GetModuleFileName$())
HelpDir$ = InstallDir$ + "lb4help\LibertyBASIC_4_web\HTML\"
ReservedWordPage$ = HelpDir$ + "libe6ldg.htm""libe6ldg.txt"
HelpFileIndex$ = InstallDir$; "lb4help\LibertyBASIC_4.html"

nomainwin
WindowWidth = 442
```

```
WindowHeight = 520
UpperLeftX   = DisplayWidth - WindowWidth
UpperLeftY   = 40

statictext #s.search "Search for: ", 10, 15, 200, 20
textbox #s.term 10, 35, 320, 25
button #s.default "Find", [getIt], UL, 340, 35, 80, 25
listbox #s.pages, found$(), [ListPages], 10, 65, 410, 200
statictext #s.heading
"Commands and Reserved Word List", 10, 270, 300, 20
graphicbox #s.reserved 10, 290, 410, 170
'container for browser control
button #s.content "Contents", [startBrowser], UL, 340, 5, 80, 25

open "Search Me!" for dialog_nf as #s
#s "trapclose [quit]"

#s.content "!font ms_sans_serif 9 bold"
#s.search "!font ms_sans_serif 9 bold"
#s.heading "!font ms_sans_serif 9 bold"

gosub [createBrowseControl]
'branch to load browser control and reserved word list

#s.term "!setfocus"

dim info$(10,10)
files HelpDir$, "*.htm", info$()

numFiles = val(info$(0,0))
wait

[getIt]
#s.term "!contents? term$"

dim found$(numFiles)

for i = 1 to numFiles

    open HelpDir$ + info$(i,0) for input as #file
        txt$ = INPUT$(#file, lof(#file))

        if instr(txt$, term$) > 0 then
            found$(i) = GetTitle$(txt$); chr$(0); info$(i,0)
            #s.pages "reload"
        end if
```

```
        close #file

    next
    wait

[ListPages]
    #s.pages "selection? FileName$"
    FileName$ = word$(FileName$, 2, chr$(0))

    run
    "rundll32.exe url.dll,FileP
    rotocolHandler "; chr$(34); HelpDir$; FileName$; chr$(34)

    wait

[startBrowser]
    run
    "rundll32.exe url.dll,FileP
    rotocolHandler "; chr$(34); HelpFileIndex$; chr$(34)
    #s.term "!setfocus"
    wait

[quit]
    close #s
    close #atl
    end

'credit: Alyce's entry.
'extract title from html text
function GetTitle$(s$)
    marker1=instr(upper$(s$), "<A NAME=")
    marker1=instr(upper$(s$), "<B>",marker1)+3
    marker2=instr(upper$(s$), "</B>", marker1)
    lenTitle=marker2-marker1
    GetTitle$=mid$(s$,marker1,lenTitle)
end function

'borrowed from Stefan Pendl
'added 5/9/11
function GetModuleFileName$()
    nSize = _MAX_PATH + 1
    lpFilename$ = space$(nSize); CHR$(0)
    callDll #kernel32, "GetModuleFileName",_
        hModule      as uLong,_
        lpFilename$ as ptr,_
        nSize        as uLong,_
```

```
        result      as uLong
    if result > 0 then GetModuleFileName$ = trim$(lpFilename$)
end function
```

```
function GetFolder$(Path$)
    pos = 1
    GetFolder$ = Path$
    while pos > 0
        pos = instr(Path$, "\", pos)
        if pos > 0 then
            GetFolder$ = left$(Path$, pos)
            pos = pos + 1
        end if
    wend
end function
```

'The following routine was copied nearly verbatim from the ATL Tutorial at

' <http://lbpe.wikispaces.com/ATL+Tutorial> by Alyce Watson. I used the first

' three snippets from the demo - ATL DLL, and Creating the control. I

' referred to the entries by Stefan and Alyce to learn what variables needed

' to be changed to fit my program. I was surprised at how well it worked for

' so little investment of my time and energy.

[createBrowseControl]

```
    hWndContainer = hWnd(#s.reserved)    'graphicbox handle
```

```
    Open "atl" For DLL As #atl
```

```
    CallDLL #atl, "AtlAxWinInit", Ret As long
```

```
    callDLL #user32, "GetWindowLongA", _
        hWndContainer as ulong, _
        _GWL_HINSTANCE as long, _
        hInst as ulong
```

```
    ResName$ = ReservedWordPage$
```

```
    style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL
```

```
    CallDLL #user32, "CreateWindowExA", _
        _WS_EX_STATICEDGE As long, _    'extended type
        "AtlAxWin" As ptr, _            'class name
```

```
        ResName$ As ptr, _
'URL, or progID or CLSID -file name
        style As long, _           'window style
        0 As long, _               'left x pos
        0 As long, _               'top y pos
        408 As long, _             'width
        168 As long, _             'height
        hWndContainer As ulong, _  'handle of container
        100 As long, _
'handle to menu or child window ID
        hInst As ulong, _          'parent instance handle
        0 As long, _               'window creation data
        hATL As ulong
'handle of active template library control

return
```

Version 1.01 added with Browser Control

```
' LB Help Search entry
' Author: jaba
' Date: May 5, 2011
'
' History:
' v1.0      May 6, 2011      Submitted
' v1.01     May 6, 2011     Added browser control
'
' This code is free for personal use.
'
' Based on a framework provided by Alyce on LB forum.
'
' Some parts taken from entries by others, including
' Alyce, Janet, Stefan, and Gordon.
'
' Thanks to them and others.
'

' This is a basic and minimum interface which uses the default browser
' to display search terms. As I learn more, I may include a browser co
ntrol

' in the interface, which I think is the best and fastest way to displ
ay
```

```
' the search terms. But, for now, I'm happy to be making a submission.
'
' To display the Reserved Word List, I edited the .htm file and saved
it
' as a .txt file in my /HTML/ sub folder. The user will see the html
' file in the lower textbox, which includes html tags. I have not made
a
' parser yet, or included the browser control to display html files.
' At this point, it's an idea only.
'

HelpDir$ = StartupDir$ + "lb4help\LibertyBASIC_4_web\HTML\"
ReservedWordPage$ = HelpDir$ + "libe6ldg.htm""libe6ldg.txt"
HelpFileIndex$ = StartupDir$; "lb4help\LibertyBASIC_4.html"

nomainwin
    WindowWidth = 442
    WindowHeight = 520
    UpperLeftX = DisplayWidth - WindowWidth
    UpperLeftY = 40

    statictext #s.search "Search for: ", 10, 15, 200, 20
    textbox #s.term 10, 35, 320, 25
    button #s.default "Find", [getIt], UL, 340, 35, 80, 25
    listbox #s.pages, found$(), [ListPages], 10, 65, 410, 200
    statictext #s.heading
"Commands and Reserved Word List", 10, 270, 300, 20
'' stylebits #s.reserved, _WS_VSCROLL, _ES_AUTOHSCROLL, 0, 0
'' textbox #s.reserved 10, 290, 410, 170
    graphicbox #s.reserved 10, 290, 410, 170
'container for browser control
    button #s.content "Contents", [startBrowser], UL, 340, 5, 80, 25

    open "Search Me!" for dialog_nf as #s
    #s "trapclose [quit]"

    #s.content "!font ms_sans_serif 9 bold"
    #s.search "!font ms_sans_serif 9 bold"
```

```
#s.heading "!font ms_sans_serif 9 bold"
'' #s.reserved "!font courier_new 8"

gosub [createBrowseControl]
'branch to load browser control and reserved word list

'' #s.reserved ReservedWordList$(ReservedWordPage$)
#s.term "!setfocus"

dim info$(10,10)
files HelpDir$, "*.htm", info$()

numFiles = val(info$(0,0))
wait

[getIt]
#s.term "!contents? term$"

dim found$(numFiles)

for i = 1 to numFiles

    open HelpDir$ + info$(i,0) for input as #file
        txt$ = INPUT$(#file, lof(#file))

        if instr(txt$, term$) > 0 then
            found$(i) = GetTitle$(txt$); chr$(0); info$(i,0)
            #s.pages "reload"
        end if
    close #file

next
wait

[ListPages]
#s.pages "selection? FileName$"
FileName$ = word$(FileName$, 2, chr$(0))

run
"rundll32.exe url.dll,FileP
rotocolHandler "; chr$(34); HelpDir$; FileName$; chr$(34)

wait

[startBrowser]
```

```
run
"rundll32.exe url.dll,FileP
rotocolHandler "; chr$(34); HelpFileIndex$; chr$(34)
  #s.term "!setfocus"
wait

[quit]
  'How can I close browser window programmatically?
  ' notice "Be sure to close browser!"
  close #s
  close #atl
end

'credit: Alyce's entry.
'extract title from html text
function GetTitle$(s$)
  marker1=instr(upper$(s$), "<A NAME=")
  marker1=instr(upper$(s$), "<B>",marker1)+3
  marker2=instr(upper$(s$), "</B>", marker1)
  lenTitle=marker2-marker1
  GetTitle$=mid$(s$,marker1,lenTitle)
end function

''This function is not needed when using browser control
function ReservedWordList$(page$)
  open page$ for input as #file
  ReservedWordList$ = INPUT$(#file, lof(#file))
  close #file
end function

''The following routine was copied nearly verbatim from the ATL Tutori
al at
'' http://lbpe.wikispaces.com/ATL+Tutorial by Alyce Watson. I used th
e first
'' three snippets from the demo - ATL DLL, and Creating the control.
I
'' refered to the entries by Stefan and Alyce to learn what variables
needed
'' to be changed to fit my program. I was surprised at how well it wo
rked for
'' so little investment of my time and energy.
[createBrowseControl]
  hWndContainer = hWnd(#s.reserved)   'graphicbox handle
```

```
Open "atl" For DLL As #atl
CallDLL #atl, "AtlAxWinInit", Ret As long

callDLL #user32, "GetWindowLongA", _
    hWndContainer as ulong, _
    _GWL_HINSTANCE as long, _
    hInst as ulong

ResName$ = ReservedWordPage$

style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL
CallDLL #user32, "CreateWindowExA", _
    _WS_EX_STATICEDGE As long, _      'extended type
    "AtlAxWin" As ptr, _              'class name
    ResName$ As ptr, _
'URL, or progID or CLSID -file name
    style As long, _                  'window style
    0 As long, _                      'left x pos
    0 As long, _                      'top y pos
    408 As long, _                    'width
    168 As long, _                    'height
    hWndContainer As ulong, _         'handle of container
    100 As long, _
'handle to menu or child window ID
    hInst As ulong, _                 'parent instance handle
    0 As long, _                     'window creation data
    hATL As ulong
'handle of active template library control
return
```

Initial submission Version 1.0

```
' LB Help Search entry
' Author: jaba
' Date: May 5, 2011
',
' History:
' v1.0 May 6, 2011 Submitted
',
',
' This code is free for personal use.
',
```

```
' Based on a framework provided by Alyce on LB forum.
',
' Some parts taken from entries by others, including
' Alyce, Janet, Stefan, and Gordon.
',
' Thanks to them and others.
',
' This is a basic and minimum interface which uses the default browse
r
' to display search terms. As I learn more, I may include a browser c
ontrol
' in the interface, which I think is the best and fastest way to disp
lay
' the search terms. But, for now, I'm happy to be making a submission
.
',
' To display the Reserved Word List, I edited the .htm file and saved
it
' as a .txt file in my /HTML/ sub folder. The user will see the html
' file in the lower textbox, which includes html tags. I have not mad
e a
' parser yet, or included the browser control to display html files.
' At this point, it's an idea only.
',
```

```
HelpDir$ = StartupDir$ + "lb4help\LibertyBASIC_4_web\HTML\"
ReservedWordPage$ = HelpDir$ + "libe6ldg.htm""libe6ldg.txt"
HelpFileIndex$ = StartupDir$; "lb4help\LibertyBASIC_4.html"
```

```
nomainwin
```

```
WindowWidth = 442
WindowHeight = 520
UpperLeftX = DisplayWidth - WindowWidth
UpperLeftY = 40
```

```
statictext #s.search "Search for: ", 10, 15, 200, 20
textbox #s.term 10, 35, 320, 25
button #s.default "Find", [getIt], UL, 340, 35, 80, 25
listbox #s.pages, found$(), [ListPages], 10, 65, 410, 200
statictext #s.heading
"Commands and Reserved Word List", 10, 270, 300, 20
stylebits #s.reserved, _WS_VSCROLL, _ES_AUTOHSCROLL, 0, 0
textbox #s.reserved 10, 290, 410, 170
button #s.content "Contents", [startBrowser], UL, 340, 5, 80, 25
```

```
open "Search Me!" for dialog_nf as #s
#s "trapclose [quit]"

#s.content "!font ms_sans_serif 9 bold"
#s.search "!font ms_sans_serif 9 bold"
#s.heading "!font ms_sans_serif 9 bold"
#s.reserved "!font courier_new 8"

#s.reserved ReservedWordList$(ReservedWordPage$)
#s.term "!setfocus"

dim info$(10,10)
files HelpDir$, "*.htm", info$()

numFiles = val(info$(0,0))
wait

[getIt]
#s.term "!contents? term$"

dim found$(numFiles)

for i = 1 to numFiles

open HelpDir$ + info$(i,0) for input as #file
txt$ = INPUT$(#file, lof(#file))

if instr(txt$, term$) >0 then
found$(i) = GetTitle$(txt$); chr$(0); info$(i,0)
#s.pages "reload"
end if
close #file

next
wait

[ListPages]
#s.pages "selection? FileName$"
FileName$ = word$(FileName$, 2, chr$(0))

run
"rundll32.exe url.dll,FileP
rotocolHandler "; chr$(34); HelpDir$; FileName$; chr$(34)
```

```
wait
```

```
[startBrowser]
```

```
run
```

```
"rundll32.exe url.dll,FileP
```

```
rotocolHandler "; chr$(34); HelpFileIndex$; chr$(34)
```

```
#s.term "!setfocus"
```

```
wait
```

```
[quit]
```

```
'How can I close browser window programmatically?
```

```
notice "Be sure to close browser!"
```

```
close #s
```

```
end
```

```
'credit: Alyce's entry.
```

```
'extract title from html text
```

```
function GetTitle$(s$)
```

```
marker1=instr(upper$(s$), "<A NAME=")
```

```
marker1=instr(upper$(s$), "<B>",marker1)+3
```

```
marker2=instr(upper$(s$), "</B>", marker1)
```

```
lenTitle=marker2-marker1
```

```
GetTitle$=mid$(s$,marker1,lenTitle)
```

```
end function
```

```
function ReservedWordList$(page$)
```

```
open page$ for input as #file
```

```
ReservedWordList$ = INPUT$(#file, lof(#file))
```

```
close #file
```

```
end function
```