

Active Template Library with Liberty BASIC

Embedding a Browser or ActiveX Control in a Liberty BASIC Window

May 1, 2006 -

[Alyce](#) (Alyce Watson)

Table of Contents

[Active Template Library with Liberty BASIC](#)

[Embedding a Browser or ActiveX Control in a Liberty BASIC Window](#)

[ATL DLL](#)

[Creating the Control](#)

[Destroying the Control](#)

[Multiple ATL Controls](#)

[Using a Graphicbox as a Container](#)

[Sizing the Graphicbox and ATL Control](#)

[Using ATL with Raw HTML](#)

[Using ATL with CLSID](#)

[Using ATL with progID](#)

[Using ATL to Embed a Browser](#)

Liberty BASIC is not able to do Component Object Model (COM) programming natively, which makes the use of ActiveX controls difficult. You can use the LB_Dispatcher ActiveX tools provided by Dennis McKinney to make use of ActiveX and COM with Liberty BASIC.

- [Using ActiveX DLLs in Liberty BASIC](#)

The Active Template Library (ATL) is a set of template-based C++ classes that simplify the programming

of Component Object Model (COM) objects. It is part of the Windows Operating System. Liberty BASIC cannot easily make use of this complete library, but there is a simple way to use a small part of it. You can do this by adding generic control containment capability to any window, so that the window can host ActiveX controls.

Please notice that this method provides very limited usefulness. It allows you to embed certain controls, a browser, or an HTML document into a Liberty BASIC Window. Your program has little or no control over the usage of the embedded control. This method is best used to do simple things, like display an animated GIF in a program window, or call up an HTML or PDF help system.

ATL DLL

This method requires the opening of the ATL.DLL that is part of the Windows Operating System. You must then initialize the DLL. The initialization function does not return a value.

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

Creating the Control

The control will be created with the CreateWindowExA API call, but to make that call, you first need the instance handle of the program window. Do this with GetWindowLongA in the following manner:

```
CallDLL #user32, "GetWindowLongA", _
hWndContainer As ulong, _           'handle of container window
_GWL_HINSTANCE As long, _           'flag to get instance handle
hInst As ulong                      'returns instance handle
```

You can now make the API call to CreateWindowExA, using any of the possible window styles and extended window styles. You must include the handle of the container window, as well as the instance handle of that window. You also need to specify the X,Y location for the control and its width and height. The important part of this call is the class name. You must use "AtlAxWin" as the class name. This class was registered when you made the call to AtlAxWinInit.

```
style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL
CallDLL #user32, "CreateWindowExA", _
_WS_EX_STATICEDGE As long, _          'extended type
"AtlAxWin" As ptr, _                  'class name
fileName$ As ptr, _                  'URL, or progID or CLSID
style As long, _                      'window style
0 As long, _                          'left x pos
```

```
0 As long, _                                'top y pos
cw As long, _                                'width
ch 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
```

Notice the fileName\$ argument. *This is very important.* The argument must be a program's progID, CLSID, or a URL. If it does not resolve to a valid progID or CLSID, it is assumed to be a URL and the MSIE web browser control is loaded. Since Internet Explorer includes the functionality to display GIFs, JPEGs, PDF files, TXT files, DOC files, HTML files, etc., your program window can contain any of these file types. You can also browse to a website. If fileName\$ is "<http://www.libertybasic.com/>" the browser is loaded and it navigates to the Liberty BASIC website.

This fileName\$ string can also be raw HTML (prefixed with "MSHTML:")

Microsoft tells us this:

"AtlAxWin" is the name of a window class that helps provide ATL's control hosting functionality. When you create an instance of this class, the window procedure will automatically use the control hosting API to create a host object associated with the window and load it with the control that you specify as the title of the window.

Destroying the Control

There is no need to destroy the control. When your program window is closed, all child windows are automatically destroyed. To create the most stable application, close the program window before closing the DLL.

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

If it is necessary to destroy the ATL control before the program ends, use the DestroyWindow API call, like this:

```
CallDLL #user32, "DestroyWindow", _
  hATL As ulong, _
```

r as long

Multiple ATL Controls

Microsoft tells us this: *It is not possible to host more than one control in a single ATL host window. Each host window is designed to hold exactly one control at a time.*

Using a Graphicbox as a Container

The example programs that follow use a graphicbox as the parent, or container, for the ATL control. This allows a program to have more than one ATL control on a window. It also allows you to use native Liberty BASIC commands to enable, disable, show and hide the ATL control, by sending those commands to the graphicbox.

Sizing the Graphicbox and ATL Control

If you want the graphicbox to fill the client area of your window, use the GetClientRect API call, as in the following examples. Use the coordinates that are placed into the Rect struct to set the size for the graphicbox. Use them also to create the ATL window.

If it is necessary to resize the graphicbox and ATL control, use the "locate" command for the graphicbox. Be sure to issue a "refresh" command to the window after locating the graphicbox. Use the MoveWindow API call to resize the ATL control. Examples of this technique are included in the demonstration programs that follow.

Using ATL with Raw HTML

```
nomainwin

WindowWidth=400
WindowHeight=300
UpperLeftX=1:UpperLeftY=1
menu #main, "&File", "E&xit", [quit]

graphicbox #main.g, 0, 0, 100, 100

Open "Liberty Basic ATL Demo" For Window_nf As #main
    #main "TrapClose [quit]"
```

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

hWndViewer = hWnd(#main.g) 'Windows handle of graphicbox
hMain = hWnd(#main) 'Windows handle of main window

STRUCT Rect,_
    'struct for storing client area rectangle
    leftX as long,_
        'upper left x
    upperY as long,_
        'upper left y
    rightX as long,_
        'lower right x
    lowerY as long
        'lower right y

calldll #user32, "GetClientRect",_
    hMain as ulong,_
        'window handle
    Rect as struct,_
        'name of struct
    r as long
        'return
cw = Rect.rightX.struct
ch = Rect.lowerY.struct

'resize graphicbox to fill client area of window
#main.g "locate 0 0 ";cw+2;" ";ch+2
#main "refresh"

CallDLL #user32, "GetWindowLongA", _
    hWndViewer As ulong,_
        'handle of graphicbox
    _GWL_HINSTANCE As long,_
        'flag to get instance handle
    hInst As ulong
'returns instance handle of graphicbox

style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL

'a minimal HTML document:
html$ =
"MSHTML:<html><head></head><body>This is an HTML document!</body></html>"
CallDLL #user32, "CreateWindowExA", _
    _WS_EX_STATICEDGE As long,_
        'extended type
    "AtlAxWin" As ptr,_
        'class name
    html$ As ptr,_
        'URL, or progID or CLSID
    style As long,_
        'window style
    0 As long,_
        'left x pos
    0 As long,_
        'top y pos
    cw As long,_
        'width
    ch As long,_
        'height
    hWndViewer As ulong,_
        'handle of parent = graphicbox
```

```
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
wait

[quit]
Close #main
close #atl      'close DLL after closing window
end
```

Using ATL with CLSID

```
'MSCAL.Calendar.7
nomainwin

WindowWidth=400
WindowHeight=300
UpperLeftX=1:UpperLeftY=1
menu #main, "&File", "E&xit", [quit]

graphicbox #main.g, 0, 0, 100, 100

Open "Liberty Basic ATL Demo" For Window_nf As #main
    #main "TrapClose [quit]"

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

    hWndViewer = hWnd(#main.g) 'Windows handle of graphicbox
    hMain = hWnd(#main)          'Windows handle of main window

    STRUCT Rect,_ 'struct for storing client area rectangle
        leftX as long,_      'upper left x
        upperY as long,_     'upper left y
        rightX as long,_    'lower right x
        lowerY as long       'lower right y

    calldll #user32,"GetClientRect",_
        hMain as ulong,_ 'window handle
        Rect as struct,_ 'name of struct
        r as long         'return
    cw = Rect.rightX.struct
```

```

ch = Rect.lowerY.struct

'resize graphicbox to fill client area of window
#main.g "locate 0 0 ";cw+2;" ";ch+2
#main "refresh"

CallDLL #user32, "GetWindowLongA", _
    hWndViewer As ulong, _           'handle of graphicbox
    _GWL_HINSTANCE As long, _       'flag to get instance handle
    hInst As ulong
'returns instance handle of graphicbox

style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL

CallDLL #user32, "CreateWindowExA", _
    _WS_EX_STATICEDGE As long, _     'extended type
    "AtlAxWin" As ptr, _            'class name
    "MSCAL.Calendar.7" As ptr, _    'URL, or progID or CLSID
    style As long, _                'window style
    0 As long, _                   'left x pos
    0 As long, _                   'top y pos
    cw As long, _                  'width
    ch As long, _                  'height
    hWndViewer As ulong, _         'handle of parent = graphicbox
    100 As long, _                'width
'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
wait

[quit]
    Close #main
    close #atl      'close DLL after closing window
    end

```

Using ATL with progID

```

'MSCAL.Calendar
nomainwin

WindowWidth=400
WindowHeight=300
UpperLeftX=1:UpperLeftY=1

```

```
menu #main, "&File", "E&xit", [quit]

graphicbox #main.g, 0, 0, 100, 100

Open "Liberty Basic ATL Demo" For Window_nf As #main
  #main "TrapClose [quit]"

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

hWndViewer = hWnd(#main.g) 'Windows handle of graphicbox
hMain = hWnd(#main)           'Windows handle of main window

STRUCT Rect,_
  'struct for storing client area rectangle
  leftX as long,_
    'upper left x
  upperY as long,_
    'upper left y
  rightX as long,_
    'lower right x
  lowerY as long
    'lower right y

calldll #user32,"GetClientRect",_
  hMain as ulong,_
    'window handle
  Rect as struct,_
    'name of struct
  r as long
    'return
cw = Rect.rightX.struct
ch = Rect.lowerY.struct

'resize graphicbox to fill client area of window
#main.g "locate 0 0 ";cw+2;" ";ch+2
#main "refresh"

CallDLL #user32, "GetWindowLongA", _
  hWndViewer As ulong,_
    'handle of graphicbox
  _GWL_HINSTANCE As long,_
    'flag to get instance handle
  hInst As ulong
'returns instance handle of graphicbox

style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL

CallDLL #user32, "CreateWindowExA", _
  _WS_EX_STATICEDGE As long,_
    'extended type
  "AtlAxWin" As ptr,_
    'class name
  "MSCAL.Calendar" As ptr,_
    'URL, or progID or CLSID
  style As long,_
    'window style
  0 As long,_
    'left x pos
  0 As long,_
    'top y pos
  cw As long,_
    'width
```

```
        ch As long, _                                'height
        hWndViewer As ulong, _                      'handle of parent = graphicbox
        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
wait

[quit]
    Close #main
    close #atl      'close DLL after closing window
    end
```

Using ATL to Embed a Browser

Two demos follow. The first is a minimal program that shows how to embed a web browser in a Liberty BASIC Window. The second demo shows how to use that embedded browser to display various types of files, and how to resize it as the user resizes the program window.

```
'the minimal browser

nomainwin

WindowWidth=DisplayWidth-100
WindowHeight=DisplayHeight-100
UpperLeftX=1:UpperLeftY=1
menu #main, "&File", "E&xit", [quit]

graphicbox #main.g, 0, 0, 100, 100

Open "Liberty Basic Browser ATL Demo" For Window_nf As #main
    #main "TrapClose [quit]"

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

    hWndViewer = hWnd(#main.g) 'Windows handle of graphicbox
    hMain = hWnd(#main)         'Windows handle of main window

    STRUCT Rect,_
        'struct for storing client area rectangle
        leftX as long,_
            'upper left x
```

```
upperY as long, _           'upper left y
rightX as long, _           'lower right x
lowerY as long              'lower right y

call dll #user32, "GetClientRect", _
    hMain as ulong, _ 'window handle
    Rect as struct, _ 'name of struct
    r as long          'return
cw = Rect.rightX.struct
ch = Rect.lowerY.struct

'resize graphicbox to fill client area of window
#main.g "locate 0 0 ";cw+2;" ";ch+2
#main "refresh"

Call DLL #user32, "GetWindowLongA", _
    hWndViewer As ulong, _           'handle of graphicbox
    _GWL_HINSTANCE As long, _       'flag to get instance handle
    hInst As ulong
'returns instance handle of graphicbox

style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL

url$ = "http://www.libertybasic.com/"
Call DLL #user32, "CreateWindowExA", _
    _WS_EX_STATICEDGE As long, _     'extended type
    "AtlAxWin" As ptr, _           'class name
    url$ As ptr, _                 'URL, or progID or CLSID
    style As long, _               'window style
    0 As long, _                   'left x pos
    0 As long, _                   'top y pos
    cw As long, _                 'width
    ch As long, _                 'height
    hWndViewer As ulong, _         'handle of parent = graphicbox
    100 As long, _                'width of window
'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
wait

[quit]
    Close #main
    close #atl      'close DLL after closing window
end
```

The browser can be used to view various file types.

```
nomainwin
hATL = 0      'handle of ATL control
cw = 0        'client area width
ch = 0        'client area height

STRUCT Rect,_ 'struct for storing client area rectangle
    leftX as long,_      'upper left x
    upperY as long,_     'upper left y
    rightX as long,_    'lower right x
    lowerY as long       'lower right y

WindowWidth=DisplayWidth-200 : UpperLeftX=1
WindowHeight=DisplayHeight-200 : UpperLeftY=1

menu #main, "&File", "&Open ATL", [openFile], "Open &URL", [openURL], _
    "&Close ATL", [closeATL], |, "E&xit", [quit]
menu #main, "&Options", "&Enable", [EnableMe], "&Disable", [
    DisableMe],_
    "&Show", [ShowMe], "&Hide", [HideMe]

graphicbox #main.g, 0, 0, 100, 100
Open "ATL Demo" For Window As #main
    #main "TrapClose [quit]"
    #main "resizehandler [resizeMe]"

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

hBox = hWnd(#main.g) 'Windows handle of graphicbox
hMain = hWnd(#main)  'Windows handle of main window

CallDLL #user32, "GetWindowLongA", _
    hBox As ulong, _           'handle of graphicbox
    _GWL_HINSTANCE As long, _  'flag to get instance handle
    hInst As ulong
'returns instance handle of graphicbox

calldll #user32, "GetClientRect",_
    hMain as ulong,_ 'window handle
    Rect as struct,_ 'name of struct
    r as long         'return
cw = Rect.rightX.struct
ch = Rect.lowerY.struct
```

```
gosub [doResize]
WAIT

[quit] 'quit from menu, or user closes window
close #main
close #atl      'close DLL after closing window
end

[resizeMe] 'activated when user resizes window
cw = WindowWidth      'LB puts client width into WindowWidth
ch = WindowHeight     'LB puts client height into WindowHeight
gosub [doResize]
wait

[doResize] 'expects client width and height to be in cw, ch
'resize graphicbox to fill client area of window
#main.g "locate 0 0 ";cw+2;" ";ch+2
#main "refresh"
if hATL then 'resize active template control if it exists
  calldll #user32, "MoveWindow",_
  hATL as ulong, _      'handle of ATL
  0 as long, 0 as long, cw as long, ch as long,_
  1 as long, result as long
end if
return

[openFile]
'ATL will use browser capability to open files of many types
  filedialog "Open file",
"*.txt;*.rtf;*.doc;*.pdf;*.jpg;*.gif;*.xls;*.html;*.htm", fileName$
  if fileName$ = "" then wait
  gosub [openATL]
  wait

[openURL] 'if given a URL, ATL embeds a browser control
  prompt "Type URL to website.";urlName$
  if urlName$ = "" then wait
  fileName$ = urlName$
  gosub [openATL]
  wait

[openATL] 'expects fileName$ to contain url or filename
  if hATL then gosub [closeFile]
  style = _WS_CHILD or _WS_VISIBLE or _WS_VSCROLL
  CallDLL #user32, "CreateWindowExA", _
    _WS_EX_STATICEDGE As long, _'extended type
```

```
"AtlAxWin" As ptr, _           'class name
fileName$ As ptr, _           'URL, or progID or CLSID
style As long, _             'window style
0 As long, _                 'left x pos
0 As long, _                 'top y pos
cw As long, _                'width
ch As long, _                'height
hBox As ulong, _             'handle of parent = graphicbox
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

[closeATL] 'user clicks menu to close file
  if hATL then gosub [closeFile]
  wait

[closeFile] 'destroy the atl child, nullify hATL variable
  CallDLL #user32, "DestroyWindow", _
    hATL As ulong, _
    r as long
  hATL = 0
  return

[EnableMe] 'menu click to enable ATL
  #main.g "enable"
  wait

[DisableMe] 'menu click to disable ATL
  #main.g "disable"
  wait

>ShowMe] 'menu click to show ATL
  #main.g "show"
  wait

[HideMe] 'menu click to hide ATL
  #main.g "hide"
  wait
```

Table of Contents

[Active Template Library with Liberty BASIC](#)

[Embedding a Browser or ActiveX Control in a Liberty BASIC Window](#)

[ATL DLL](#)

[Creating the Control](#)

[Destroying the Control](#)

[Multiple ATL Controls](#)

[Using a Graphicbox as a Container](#)

[Sizing the Graphicbox and ATL Control](#)

[Using ATL with Raw HTML](#)

[Using ATL with CLSID](#)

[Using ATL with progID](#)

[Using ATL to Embed a Browser](#)