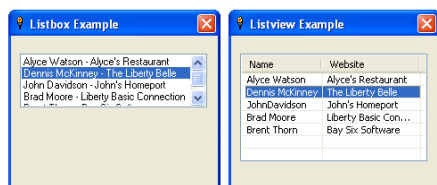


## Trapping a Double Click in a Listview

Demo by Eldron Gill (- [eaglesoar](#))

### Editor's Note

A [listview](#) is an advanced [listbox](#). The listview control is typically used to display members of a data collection in table format. Each column has its own column header. The listview is far more complex than the ordinary listbox. While the listbox is easily populated from its associated array, the listview requires each item to be individually assigned. This assignment is accomplished with a rather complicated series of structs and API calls.



A listview demo is included in Alyce Watson's (- [Alyce](#)) [Liberty 4 Companion ebook](#). Alyce's demo shows how to construct a listview, add and delete items to the listview, and make a listview selection by highlighting an item and then clicking the select button. Eldron Gill takes Alyce's demo one step further by allowing selection from a double mouse click on the listview item itself.

Capturing mouse clicks in a listview requires a callback function, not easily handled even with Liberty BASIC. Fortunately, Dennis McKinney (- [DennisMcK](#)) offers the free [msghook.dll](#) designed specifically for this purpose. Here is Eldron's modification of Alyce's listview demo to allow selection by a double left mouse click. As stated in the code, [msghook.dll](#) must be accessible within the program folder.

It should be mentioned that Eldron sought and received permission from both Alyce and Dennis before submitting this demo. Our thanks to all three for this collaboration and sharing of resources and knowledge.

```
'This demo requires msghook.dll by Dennis McKinney
'Get it here: http://www.syberden.net/libertybelle/dlls.htm
'Alyce Watson provided the listview code.
'http://www.alyclesrestaurant.com
```

```
'listview control with double click 10/09/2006 Eldron Gill
```

```
nomainwin
```

```
dim info$(0,0)
If fileExists(DefaultDir$, "msghook.dll") < 1 then
```

```
    notice ""+chr$(13)+
"This demo requires msghook.dll by Dennis McKinney"+chr$(13)+_

"Download the dll from here : http://www.syberden.net/libertybelle/dlls.htm"+chr$(13)+_
    "Place it in the folder with this code."
End
End If

'constants
LVS.NOSORTHEADER = 32768
LVS.REPORT = 1
LVS.SINGLESEL = 4
LVS.SHOWSELALWAYS = 8
LVS.SORTASCENDING = 16
LVS.SORTDESCENDING = 32
LVS.NOLABELWRAP = 128
LVS.AUTOARRANGE = 256
LVS.NOSCROLL = 8192
LVS.ALIGNTOP = 0
LVS.ALIGNLEFT = 2048
LVS.NOCOLUMNHEADER = 16384
LVIF.TEXT = 1
LVIF.STATE = 8
LVIS.UNSELECTED = 0
LVIS.FOCUSED = 1
LVIS.SELECTED = 2
LVM.FIRST = 4096
LVM.SETITEM = 4102
LVM.INSERTITEM = 4103
LVM.INSERTCOLUMN = 4123
LVM.GETITEMCOUNT = 4100
LVM.GETITEMA = 4101
LVM.GETITEMTEXTA = 4141
LVM.GETITEMSTATE = 4138
LVM.SETITEMSTATE = 4139
LVM.DELETEITEM = 4104
LVM.DELETEALLITEMS = 4105
LVCF.WIDTH = 2
LVCF.TEXT = 4

'create structs
Struct LVCOLUMN, _
    mask As ulong, _
    fmt As long, _
    cx As long, _
```

```
    pszText$ As ptr, _
    cchTextMax As long, _
    iSubItem As long, _
    iImage As long, _
    iOrder As long

Struct LVITEM, _
    mask As ulong, _
    iItem As long, _
    iSubItem As long, _
    state As ulong, _
    stateMask As ulong, _
    pszText$ As ptr, _
    cchTextMax As long, _
    iImage As long, _
    lParam As long, _
    iIndent As long

struct msg,_
    hndl as ulong,_
    message as long,_
    wParam as long,_
    lParam as long,_
    LOWORDwparam as word,_
    HIWORDwparam as word,_
    LOWORDlparam as word,_
    HIWORDlparam as word

'initialize common controls:

call dll #comctl32, "InitCommonControls",re as void

' Open a window
WindowWidth = 240: WindowHeight = 200
UpperLeftX = 10: UpperLeftY = 10
button #1.b, "Add",[add],UL,10,10,90,24
button #1.d, "Delete",[delete],UL,120,10,90,24
button #1.GetMsgHookCallback, "", [choice], ul, 0, 0, 0, 0
statictext #1.s, "List count: 2",10,140,100,30
open "Listview Example" for dialog as #1

print #1, "trapclose [quit]"

    hwndParent = hwnd(#1)
```

```
' Get window instance handle
CallDLL #user32, "GetWindowLongA", _
    hwndParent As ulong, _           'parent window handle
    _GWL_HINSTANCE As long, _       'flag to retrieve instance handle
    hInstance As ulong              'instance handle

' Create control
    style = _WS_CHILD OR _WS_VISIBLE OR LVS.NOSORTHEADER _
    OR LVS.REPORT OR LVS.SINGLESEL OR LVS.SHOWSELALWAYS

calldll #user32, "CreateWindowExA", _
    _WS_EX_CLIENTEDGE As long, _    ' extended style
    "SysListView32" as ptr, _       ' class name
    "" as ptr, _
    style as long, _                ' style
    10 as long, _                   ' left x
    50 as long, _                   ' top y
    200 as long, _                  ' width
    80 as long, _                   ' height
    hwndParent as ulong, _          ' parent hWnd
    0 as long, _
    hInstance as ulong, _           ' hInstance
    "" as ptr, _
    hwndLV as long                  ' listview handle

'insert first column:
LVCOLUMN.mask.struct = LVCF.WIDTH OR LVCF.TEXT
LVCOLUMN.cx.struct = 90
LVCOLUMN.pszText$.struct = "Name"
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.INSERTCOLUMN As long, _
    0 As long, _                   '0 = first column
    LVCOLUMN As struct, _
    r As long

'insert second column:
LVCOLUMN.cx.struct = 65
LVCOLUMN.pszText$.struct = "Rank"
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.INSERTCOLUMN As long, _
    1 As long, _                   '1 = second column
    LVCOLUMN As struct, _
    r As long

'insert text for first row, first column
'requires message to insert item
LVITEM.mask.struct = LVIF.TEXT
LVITEM.iItem.struct = 0           'first row
```

```
LVITEM.iSubItem.struct = 0 'first column
LVITEM.pszText$.struct = "Carl Gundel"
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.INSERTITEM As long, _
    0 As long, _
    LVITEM As struct, _
    r As long
'insert text for second column, first row
LVITEM.iItem.struct = 0 'first row
LVITEM.iSubItem.struct = 1 'second column
LVITEM.pszText$.struct = "Expert"
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.SETITEM As long, _ '
    0 As long, _
    LVITEM As struct, _
    r As long
'insert second row, first column
LVITEM.iItem.struct = 1 'second row
LVITEM.iSubItem.struct = 0 'first column
LVITEM.pszText$.struct = "Bill Gates"
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.INSERTITEM As long, _
    0 As long, _
    LVITEM As struct, _
    r As long
'add second column to second row
LVITEM.iItem.struct = 1 'second row
LVITEM.iSubItem.struct = 1 'second column
LVITEM.pszText$.struct = "Novice"
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.SETITEM As long, _ '
    0 As long, _
    LVITEM As struct, _
    r As long

'full row select
LVM.FIRST = hexdec("1000")
LVM.SETEXTENDEDLISTVIEWSTYLE = LVM.FIRST + 54
LVS.EX.FULLROWSELECT = hexdec("20")

CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
```

```
LVM.SETEXTENDEDLISTVIEWSTYLE As long, _ '
LVS.EX.FULLROWSELECT As long, _
LVS.EX.FULLROWSELECT As long, _
r As long

' if you want a grid, leave next message uncommented...
CallDll #user32, "SendMessageA" , hwndLV as
uLong, 4150 as Long,_
1 As Long, 1 As Long, re as Long '

open "MsgHook" for dll as #MsgHook

hMsgProc = hwnd(#1.GetMsgHookCallback)
calldll #MsgHook, "TrapMsgFor",hwndLV as ulong, ret as long
calldll #MsgHook, "WatchMsg", hwndLV as
ulong, _WM_LBUTTONDOWNCLK as long, ret as long
hMsgProc = hwnd(#1.GetMsgHookCallback)
calldll #user32, "GetWindowLongA",hMsgProc as
ulong,_GWL_ID as short,callbackID as long
calldll #MsgHook, "CreateGetMsgProcHook", hwndParent as
ulong, callbackID as long, _
hMsgProc as ulong, hHook as ulong

wait

[choice] 'determine user selection

calldll #MsgHook, "GetMsg", msg as struct, ret as void
'get number of items in list:

CallDLL #user32, "SendMessageA", _
hwndLV As ulong, _
LVM.GETITEMCOUNT As long, _
0 As long, _ 'always 0
0 As long, _ 'always 0
total As long
for index = 0 to total-1 'check each row
LVITEM.mask.struct = LVIF.TEXT OR LVIF.STATE
LVITEM.iItem.struct = index 'row
LVITEM.iSubItem.struct = 0 'first column
LVITEM.cchTextMax.struct = 32
LVITEM.pszText$.struct = space$(32)
LVITEM.stateMask.struct = LVIS.SELECTED
CallDLL #user32, "SendMessageA", _
hwndLV As ulong, _
LVM.GETITEMA As long, _
```

```
        index As long, _          'index of row
        LVITEM As struct, _
        r As long
    state = LVITEM.state.struct 'selected state of item
    if state and LVIS.SELECTED then
        txt$=winsting(LVITEM.pszText$.struct)
        notice "Selected: ";txt$
        exit for
    end if
next
if txt$="" then notice "No selection."
txt$=""
wait
[add]
'make sure no item is in selected state:
    LVITEM.stateMask.struct = LVIS.SELECTED 'bit to set
    LVITEM.state.struct = LVIS.UNSELECTED
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.SETITEMSTATE As long, _
    -1 As long, _ 'change applies to all items
    LVITEM As struct, _
    r As long
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.GETITEMCOUNT As long, _
    0 As long, _ 'always 0
    0 As long, _ 'always 0
    count As long
name$="No Name"
prompt "Name?";name$
if name$="" then name$="No Name"
'insert next row, first column
LVITEM.mask.struct = LVIF.TEXT
LVITEM.iItem.struct = count 'next row
LVITEM.iSubItem.struct = 0 'first column
LVITEM.pszText$.struct = name$
CallDLL #user32, "SendMessageA", _
    hwndLV As ulong, _
    LVM.INSERTITEM As long, _
    0 As long, _
    LVITEM As struct, _
    r As long
level$="No Level"
prompt "Level?";level$
if level$="" then level$="No Level"
```

```
'add second column to row
    LVITEM.iItem.struct = count 'next row
    LVITEM.iSubItem.struct = 1 'second column
    LVITEM.pszText$.struct = level$
    CallDLL #user32, "SendMessageA", _
        hwndLV As ulong, _
        LVM.SETITEM As long, _
        0 As long, _
        LVITEM As struct, _
        r As long
    print #1.s, "List count: ";count+1
wait

[delete]'get user selection and delete
    'get number of items in list:
    CallDLL #user32, "SendMessageA", _
        hwndLV As ulong, _
        LVM.GETITEMCOUNT As long, _
        0 As long, _ 'always 0
        0 As long, _ 'always 0
        total As long
    for index = 0 to total-1 'check each row
        LVITEM.mask.struct = LVIF.STATE
        LVITEM.iItem.struct = index 'row
        LVITEM.iSubItem.struct = 0 'first column
        LVITEM.stateMask.struct = LVIS.SELECTED
        CallDLL #user32, "SendMessageA", _
            hwndLV As ulong, _
            LVM.GETITEM As long, _
            index As long, _ 'index of row
            LVITEM As struct, _
            r As long
        state = LVITEM.state.struct 'selected state of item
        if state and LVIS.SELECTED then
            CallDLL #user32, "SendMessageA", _
                hwndLV As ulong, _
                LVM.DELETEITEM As long, _
                index As long, _
                LVITEM As struct, _
                r As long
        exit for
    end if
next
print #1.s, "List count: ";total-1
wait
```



```
[quit]
calldll #MsgHook, "UnhookMsgHook", hHook as ulong, ret as void

close #MsgHook
calldll #user32, "DestroyWindow", _
    hwndLV as ulong, re as long
' Close handles.
close #1:end

'Function to determine if a file exists
function fileExists(path$, filename$)
    files path$, filename$, info$()
    fileExists = val(info$(0, 0)) 'non zero is true
end function
```