

There have been several folks struggling with how to manage a dual listbox style application where items from a master list are moved to a destination list. I wrote some code to do that many years ago. This is an updated version with comments. I am hoping folks can follow along. When running the program looks like this:



Here is the code (while I hate to admit that I make mistakes, this did have an error handling the no item selected to delete condition - it is now fixed):

```
'Friuty List - an example of how to manage a listbox as a destination
list building
' the list from a master listbox. This would work with comboboxes al
so.
'by Brad Moore - placed into the Public Domain Jan 2006
nomainwin
'this is how many entries the destination list box can hold
maxEntries = 100
'create the arrays
DIM arrayOne$(12)
DIM arrayTwo$(maxEntries)
'this is the master data - you can do this with read/data statements a
lso,
' wondering how this works? - check out the WORD$ function in the hel
pfile.
a$ =
"Apples Bananas Cherries Figs Grapefruit Honeydew Kiwi Lime Mango Nect
arine Papaya Pear"
FOR x = 1 TO 12 : arrayOne$(x) = word$(a$,x) : NEXT x
'initialize our counter for items in the destination listbox
idx = 0
'setup the window and controls
WindowWidth = 280
WindowHeight = 260
UpperLeftX = INT((DisplayWidth/2)-(WindowWidth/2))
UpperLeftY = INT((DisplayHeight/2)-(WindowHeight/2))
listbox #main.lbx1,arrayOne$(),WAIT,10,10,100,170
listbox #main.lbx2,arrayTwo$(),WAIT,158,10,100,170
```

```
button #main.moveBtn, ">", [move], UL, 114, 30, 40, 30
button #main.deleteBtn, "Delete", [delete], UL, 114, 62, 40, 30
button #main.deleteBtn, "Sort", [sort], UL, 114, 94, 40, 30
button #main.closeBtn, "Close", [quit], UL, 158, 189, 100, 30
button #main.inspectBtn, "Inspect Array", [inspect], UL, 52, 189, 100, 30
'open the window
OPEN "Friuty List" FOR dialog AS #main
PRINT #main, "trapclose [quit]"
PRINT #main.lbx1, "selectindex 1"
WAIT
[move]
'get the index value for the item in the master list
PRINT #main.lbx1, "selectionindex? i"
'make sure there is room to add another item
IF idx = maxEntries THEN
    notice
    "The destination list already contains the maximum number of entries"
    WAIT
END IF
'increment the index and add the item
idx = idx + 1
arrayTwo$(idx) = arrayOne$(i)
'reload causes the listbox view to be refreshed so our item shows up
PRINT #main.lbx2, "reload"
WAIT
[delete]
'get the index value for the item in the destination list
PRINT #main.lbx2, "selectionindex? i"
if i = 0 then
    'turns out the user has not selected anything to delete
    notice
    "You must select an item from the destination list to delete " + _
        "before a delete can be preformed."
    wait
end if
'if this is not the last item we need to shift the array up to fill th
e gap
IF i < idx THEN
    FOR x = i TO idx - 1
        arrayTwo$(x) = arrayTwo$(x+1)
    NEXT x
END IF
'set the last item to a null string
arrayTwo$(idx) = ""
'decrement the total number of items in the destination array
idx = idx - 1
```

```
'reload causes the listbox view to be refreshed so our item shows up
PRINT #main.lbx2,"reload"
WAIT

[inspect]
'inspect opens a new window in dialog modal form - set up that window
WindowWidth = 372
WindowHeight = 470
UpperLeftX = INT((DisplayWidth-(2 * WindowWidth))/2)
UpperLeftY = INT((DisplayHeight-(2 * WindowHeight))/2)
button #inspect.closeBtn, "Close", [closeMe], UL, 225, 395, 125, 30
texteditor #inspect.tedit, 9, 13, 340, 375
'open the window
OPEN "Array Inspector" FOR Dialog_modal AS #inspect
#inspect "trapclose [closeMe]"
#inspect "font ms_sans_serif 10"
'initialize the output string
IF idx = 0 THEN
    OUT$ = "There are no items in the destination array."
ELSE
    OUT$ = "There are " + STR$(idx) +
" items in the destination array." + CHR$(10)
END IF
'add the array items to the output string. Note the right$ trick whic
h pads index with leading zeros
FOR x = 1 TO idx
    OUT$ = OUT$ + "Item " + RIGHT$("000" + STR$(x),3) + " = " +
arrayTwo$(x) + CHR$(10)
NEXT x
'dump the output string to the texteditor
#inspect.tedit OUT$
WAIT
[closeMe]
CLOSE #inspect
WAIT

[sort]
'if there is more than one item in the destination array call the sort
function to sort it
IF idx > 1 THEN sort arrayTwo$(), 1, idx
'reload causes the listbox view to be refreshed so our item shows up
PRINT #main.lbx2,"reload"
WAIT
[quit]
CLOSE #main
END
```

- [bradmoore](#)