Getting Started with Dialog Designer for ArcView 3.X

This summary will give some code snippets of various tricks for Dialog Designer. I will show how to “clean up” a dialog when it is closed, select all elements and clear all selected elements in a listbox, search a list when text is being typed and present that list to the user, and force upper or lower case data entry. Make sure you get the ArcView project file that has working examples of these things.

Clean Up on Close:
It is a good habit to empty controls and to clear object tags when closing dialogs. If you do not do this the lists in the list box and the objects on the object tags will be saved with the project. This can cause the .apr file to grow in size and other potential errors. Here is the script I use:

```
'+++++++++++++++++ Util.CloseDialog ++++++++++++++++
'
'runs when the dialog closes, empties the list and combo boxes

‘get the list of controls
lstControls = SELF.FindbyClass(Control)

‘iterate through the controls
for each ctl in lstControls
‘clear the object tags
 ctl.SetObjectTag("")
‘empty, emptiable controls
 if ((ctl.Is(ListBox)) or (ctl.Is(ComboBox)) or (ctl.Is(TextLine))) then
  ctl.Empty
 end
end
```

Select all and clear all selected elements in a listbox.

Select all elements, Click event:

```
'++++++++++ LISTDEMO.lbt_SelAll.Click ++++++++++
'get the list box
lbxStNames = SELF.GetDialog.FindByName("lbx_StNames")

'go to the first row of the list box
lbxStNames.GoRow(0)

'select the current row
lbxStNames.SelectCurrent(true)

'go down a row at a time, selecting the current row as you go
while (lbxStNames.GoDown)
 lbxStNames.SelectCurrent(true)
end

‘run the select script to broadcast the update event
lbxStNames.Select
```

Unselect all, Click Event:

```
'++++++++++ LISTDEMO.lbt_ClearAll.Click ++++++++++

'get the list box
lbxStNames = SELF.GetDialog.FindByName("lbx_StNames")

'set the selection range to an empty rectangle, that clears the
'the list box
lbxStNames.SetSelection(Rect.MakeEmpty, false)

‘run the select script to broadcast the update event
lbxStNames.Select
```

Search a list when text is being typed and present that list to the user.
This is an example of two controls working together, a TextLine and a ComboBox. You must set the Tag property of each control to the name of the other control. Example the Tag property of the TextLine is set to “ComboBox1”, the name of the combo box. This also uses the Changed event of the TextLine. Each time a letter is typed in the TextLine, the Changed event is triggered and the script is run. The script searches a list and using a Pattern object, finds the elements of the list that match what is being typed. The new list is put in the combo box. Here is the code.

TextLine Code, Changed event:

```
'++++++++++++++ LISTDEMO.txl_StName.Changed +++++++++++
'runs when there is a change in the text line streetname
'
'the tag is set to the name of the corresponding combo box,
'containing the list of street names.
'
'written 9/98

'get the combo box
cbxStNames = SELF.GetDialog.FindByName(SELF.GetTag)

'make the combo box visible
cbxStNames.SetVisible(true)

'get the text that is being entered
strInput = SELF.GetText
patInput = Pattern.Make(strInput+"*")

'get the full list from the object tag of the form
lstTotal = SELF.GetDialog.GetObjectTag

'make an emtpy subset list
lstSubSet = {}

'iterate through the full list of street names
for each el in lstTotal
```

```
  if (el.AsString = patInput) then

  'add the value to the subset list
  lstSubSet.Add(el.AsString)

 end
end

'populate the combo box using the subset list
cbxStNames.DefineFromList(lstSubSet)
```

ComboBox Code, Select Event:

```
'+++++++++ LISTDEMO.cbx_StNames.Select +++++++++

'get the text line control
txlStName = SELF.GetDialog.FindByName(SELF.GetTag)

'get the selection from the combo box
strSelected = SELF.GetSelection

'set the textline text to the selected text from the combo box
txlStName.SetText(strSelected)

'set the focus back to the textline
txlStName.Focus
```

Force upper or lower case data entry, Changed event:

```
'+++++++++++++ LISTDEMO.txl_Case.Changed

 'change what is being typed to upper case
 SELF.SetText(SELF.GetText.Ucase)

‘change what is being typed to lower case
‘SELF.SetText(SELF.GetText.Lcase)

'position the cursor to the front of the text
SELF.SetPosition(-1)
```

If you have any questions about this document, feel free to contact me:

Robert Meier, GIS Analyst
City of Overland Park, Kanas
8500 Santa Fe Dr.
Overland Park, KS 66212
913/895-6203
rmeier@opkansas.org